

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENEROVÁNÍ A DETEKCE BAREVNÝCH MARKERŮ PRO ROZŠÍŘENOU REALITU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. PETER BEŤKO

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENEROVÁNÍ A DETEKCE BAREVNÝCH MARKERŮ PRO ROZŠÍŘENOU REALITU

SYNTHESIS AND DETECTION OF COLOR MARKERS FOR AUGMENTED REALITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PETER BEŤKO

VEDOUcí PRÁCE

SUPERVISOR

Doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2013

Abstrakt

Tato diplomová práce navazuje na výzkum v oblasti použití uniformních polí markerů pro rozšířenou realitu a doplňuje ho o možnost generování barevných polí markerů založených na obrazové předloze. Práce popisuje základy rozšířené reality a detailně vysvětluje postupy z článku *Uniform Marker Fields*. Dále je představen program na generování barevných polí markerů a nástroj na jejich detekci a vyhodnocení úspěšnosti. Práce dále prezentuje studii stálosti barev v procesu tisku a snímání. Tato studie byla vytvořena nad rámec zadání práce a získané poznatky mohou být použity i v obecnějších aplikacích využívajících barvu na kódování informace. Na závěr je popsáno začlenění poznatků do programu na generování polí markerů a z toho plynoucí zvýšení úspěšnosti detekce.

Abstract

This master's thesis emerges from the research in the field of uniform marker fields for augmented reality and broadens it by the possibility of generating colored marker fields based on arbitrary pictures. The thesis describes basics of augmented reality and explains the ideas from the *Uniform Marker Fields* paper. Presented is a program for generating colored marker fields as well as a tool for detecting markers and evaluating success rate of the detection. In addition, the work proposes a study of color constancy in the process of printing and recording. This study exceeds the scope of the requirements for this master's thesis; the knowledge can be used in arbitrary applications that use colors to code information. Finally, the integration of the knowledge into the marker field generating algorithm and resulting improvement of detection success rate are discussed.

Klíčová slova

rozšířená realita, AR, marker, Uniform Marker Fields, stálost barev

Keywords

augmented reality, AR, marker, Uniform Marker Fields, color constancy

Citace

Peter Beřko: Generování a detekce barevných markerů pro rozšířenou realitu, diplomová práce, Brno, FIT VUT v Brně, 2013

Generování a detekce barevných markerů pro rozšířenou realitu

Prohlášení

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pána Doc. Ing. Adama Herouta, Ph.D.

.....

Peter Beřko
22. května 2013

Poděkování

Ďakujem pánovi Doc. Ing. Adamovi Heroutovi, Ph.D. za odbornú pomoc a cenné rady. Veľká vďaka patrí mojim rodičom za to, že mi po každej stránke umožnili robiť to, čo ma baví.

© Peter Beřko, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Rozšírená realita	3
2.1	Rozšírená realita a jej využitie	3
2.2	Základné princípy rozšírenej reality	4
2.3	Projekty zamerané na rozšírenú realitu	6
3	Uniform Marker Fields	8
3.1	Základná myšlienka Uniform Marker Fields	8
3.2	Generovanie poľa markerov	9
3.3	Detekcia markerov a experimentálne výsledky	11
3.4	Rozšírenia Uniform Marker Fields	12
4	Implementácia algoritmov na generovanie a detekciu polí markerov	13
4.1	Použité technológie	13
4.2	Program na generovanie polí markerov	14
4.3	Program na testovanie úspešnosti detekcie	20
5	Vyhodnotenie úspešnosti detekcie a štúdie stálosti farieb	21
5.1	Úspešnosť detekcie prechodov a markerov	21
5.2	Štúdia stálosti farieb v procese tlače a snímania	25
5.3	Štúdia úspešnosti detekcie prechodov u dvojíc farieb	31
5.4	Využitie získaných poznatkov na zvýšenie úspešnosti detekcie	35
6	Záver	38
A	Adresárová štruktúra DVD	41
B	Ovládanie programov	42
B.1	Program MFGenerator	42
B.2	Program gridDetector	43
B.3	Program transitionDetector	43
B.4	Program colourTester	44

Kapitola 1

Úvod

Od prvej konferencie o rozšírenej realite s názvom *International Workshop on Augmented Reality* uplynulo 15 rokov. Za túto dobu došlo k posunu tejto oblasti od nie veľmi úspešných aplikácií demonštrujúcich jej možnosti až k robustným riešeniam fungujúcim na dnešných mobilných zariadeniach, či k pokusom o integrovanie prvkov rozšírenej reality do predmetov dennej potreby.

V súčasnosti prebieha na *Fakulte informačných technológií Vysokého učení technického* výskum v oblasti generovania a detekcie uniformných polí markerov pre aplikácie rozšírenej reality. Táto diplomová práca naň nadväzuje a rozširuje ho o možnosť generovania farebných polí markerov na základe obrazovej predlohy.

Druhá kapitola práce sa zaoberá teóriou rozšírenej reality. Podáva jej stručnú definíciu a podrobne vysvetľuje jej základné princípy. Zaoberá sa algoritmami na výpočet pozície kamery a podáva podrobný prehľad typov markerov, ktoré sú na tento účel používané.

Tretia kapitola sa zaoberá rozborom článku *Uniform Marker Fields* [6]. Je popísaná hlavná idea a výhody použitia uniformného poľa markerov, podrobne je vysvetlený postup generovania takýchto polí. Stručný popis je venovaný detekčnému algoritmu a predstavené sú aj dosiahnuté výsledky. Na záver sú spomenuté možné vylepšenia tohto prístupu, ktoré slúžia ako východiskový bod pre praktickú časť práce.

Vo štvrtej kapitole je popísaná implementácia programu na generovanie farebných polí markerov a nástroja na ich detekciu a vyhodnocovanie jej úspešnosti.

Piata kapitola začína popisom testovania. Nakoľko úspešnosť detekcie nebola uspokojivá, bolo po dohode s vedúcim práce rozhodnuté zamerať sa na odhaľovanie príčin nízkej úspešnosti a na jej zvyšovanie. Kapitola pokračuje rozsiahlou štúdiou stálosti farieb v procese tlače a snímania. Táto štúdia presahuje rámec zadania práce, získané poznatky môžu byť využité aj mimo oblasť rozšírenej reality. Na záver kapitoly je diskutované začlenenie poznatkov do algoritmu na generovanie polí markerov a z toho plynúce zvýšenie úspešnosti detekcie.

Táto práca nadväzuje na semestrálny projekt, kapitoly 2 a 3 boli z neho prebraté.

Kapitola 2

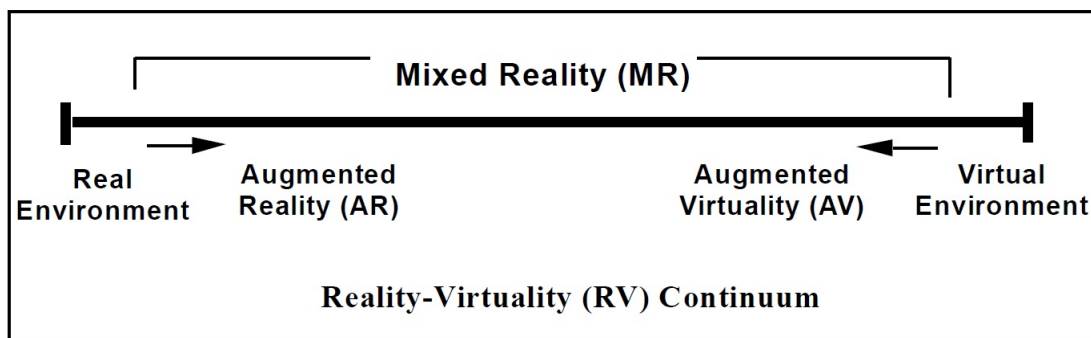
Rozšířená realita

Táto kapitola pojednáva o rozšírenej realite, ktorú definuje a uvádza jej základné princípy. Tiež predstavuje dva frameworky na tvorbu aplikácií rozšírenej reality a dva komerčne vyvíjané projekty.

2.1 Rozšířená realita a jej využitie

Pojem rozšířená realita (anglicky *Augmented Reality*) je možné definovať ako technológiu, ktorá umožňuje počítačom generovaným grafickým objektom presne prekryvať objekty fyzického sveta a to v reálnom čase [15]. Podľa prieskumu Ronalda T. Azuma [1] musí rozšířená realita spĺňať tieto tri základné podmienky: kombinovať reálne a virtuálne prostredie, byť interaktívna v reálnom čase a vkladať virtuálne objekty do reálneho sveta.

Paul Milgram a Fumio Kishino vo svojej publikácii z roku 1994 [3] umiestnili rozšírenú realitu v kontinuu reality-virtuality medzi reálne a virtuálne prostredie, teda medzi prostredie tvorené výlučne objektmi reálneho sveta a umelo vytvorené prostredie. Na obrázku 2.1 je tento priestor v kontinuu nazvaný ako zmiešaná realita (anglicky *Mixed Reality*) a tvorí ho práve rozšířená realita a rozšířená virtualita. Posledne menovaná sa od rozšírenej reality odlišuje tým, že prevažná časť scény je tvorená virtuálnymi objektmi, ktoré sú dopĺňané predmetmi z reálneho sveta. V súčasnosti sa pojem rozšířená virtualita takmer nepoužíva, zatiaľ čo pojmy rozšířená a zmiešaná realita sa často používajú ako synonymá.



Obrázok 2.1: Kontinuum reality-virtuality, prebraté z článku *Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum* [4]

Rozšířená realita sa nemusí vzťahovať iba k obrazu, v širšom kontexte je možné tento

pojem použiť aj na popis rozširovania reality o podnety vnímané ostatnými zmyslami, ako sluch, chuť, čuch, hmat.

2.2 Základné princípy rozšírenej reality

Podľa článku pána Zhou a kolegov [15], ktorý popisuje vývoj algoritmov počas desiatich rokov od prvej konferencie o rozšírenej realite, *International Workshop on Augmented Reality '98 (IWAR 98)*, je potrebné zaoberať sa týmito šiestimi aspektmi pri tvorbe aplikácií rozšírenej reality:

- hardware a software schopný vytvoriť virtuálny obraz, ktorý bude vkladáný do reálnej scény
- techniky na sledovanie zmeny pozície pozorovateľa (reprezentovaného najčastejšie kamerou, prípadne inými senzormi)
- kalibrácia zariadenia reprezentujúceho pozíciu pozorovateľa v scéne
- zobrazujúci hardware schopný spojiť virtuálny obraz s reálnym
- výpočtový hardware uspokojený na vykonávanie simulačného kódu rozšírenej reality a podporujúci vstupné a výstupné zariadenia
- techniky interakcie definujúce ako môže užívateľ interagovať s virtuálnym obsahom

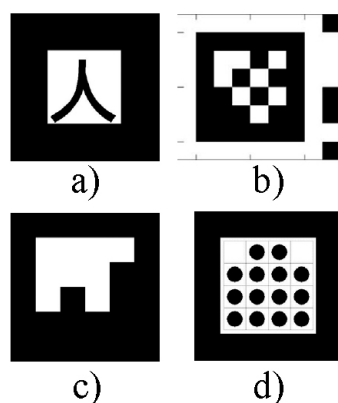
V tomto článku [15] bola tiež porovnaná dôležitosť jednotlivých aspektov a to na základe počtu výskumných publikácií, ktoré sa nimi zaoberajú. Najviac, až jedna pätina článkov, sa týkala techník na sledovanie zmeny pozície pozorovateľa. Keďže táto problematika je pre túto diplomovú prácu zásadná, bude ďalej podrobnejšie vysvetlená. Kde nie je špecifikované inak, informácie boli čerpané z publikácie pána Zhou a kolegov [15].

Prvú kategóriu tvoria techniky výpočtu pozície pozorovateľa na základe dát zo senzorov — magnetických, akustických, inerciálnych, optických alebo mechanických. Každý typ senzorov so sebou prináša špecifické výhody a nevýhody, napríklad magnetické senzory majú vysokú rýchlosť obnovovania informácie, na druhú stranu môžu byť rušené magnetickým poľom iných objektov. Jednou z možností ako zvýšiť robustnosť výpočtu pozície pozorovateľa je skombinovať rozdielne typy senzorov do jedného riešenia.

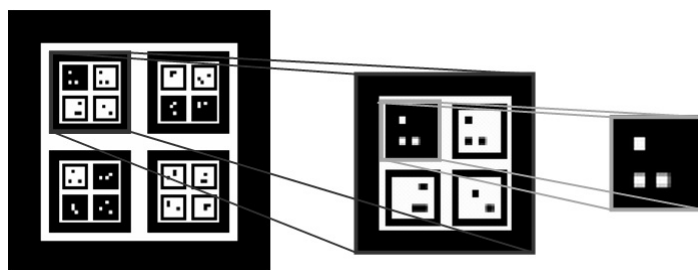
Druhou kategóriou sú techniky založené na počítačovom videní, ktoré môžeme ďalej rozdeliť na tie, ktoré sú založené na príznakoch (anglicky *feature-based*) a tie ktoré sú založené na modeli (anglicky *model-based*). Prvé z nich si kladú za cieľ nájsť korešpondenciu medzi príznakmi v 2D obraze zaznamenanom kamerou a ich 3D súradnicami v reálnom svete. Veľmi rozšíreným je využitie tzv. markerov čiže určitých značiek, ktoré sa vložia do reálnej scény a následne sú detekované algoritmom rozšírenej reality. Najčastejšie používané sú pravouhlé markery. Ich výhodou je, že obsahujú minimálne štyri rovinné body, vďaka čomu je jeden marker dostačujúci na výpočet pozície kamery. Príklad takýchto markerov je na obrázku 2.2. Inou možnosťou je použitie nepravouhlých, napríklad kruhových, markerov. V prípade že algoritmus pracuje iba so stredmi týchto kruhov (nie s kruhmi samotnými), je z každého markeru získaný iba jeden planárny bod, a teda sú potrebné minimálne tri takéto markery na výpočet pozície kamery. V prípade potreby pracovať s rozšírenou realitou vo väčšej scéne nájdeme využitie tzv. vnorené markery (anglicky *Nested markers*). Tie fungujú na princípe rekurzívnych úrovní obsahujúcich markery rôznych veľkostí. Vďaka tomu

je možné takýto vnorený marker správne detekovať ako pri jeho malej tak aj veľkej vzdialenosti od kamery. Vnorený marker je zobrazený na obrázku 2.3. V súčasnosti sa vyvíjajú pokročilé formy markerov ako napríklad *Uniform Marker Fields* (diskutované v ďalšej kapitole). Projekt *Vuforia* počíta pozíciu kamery na základe značiek, ktoré sú tvorené klasickými fotografiami.

Je tiež možné vytvoriť aplikáciu rozšírenej reality, ktorá na svoju činnosť nebude potrebovať markery. Príkladom môže byť práca J. Parka a kolegov [5], ktorí na základe známych vizuálnych príznakov vypočítajú prvotnú pozíciu kamery a následne dynamicky získavajú ďalšie príznaky z význačných bodov, čiar, či textúr, ktoré využívajú na sledovanie zmien pozície kamery.



Obrázok 2.2: Pravouhlé markery používané rôznymi systémami a inštitúciami: a) ArTool-Kit; b) Hoffman; c) Institut Graphische Datenverarbeitung; d) Siemens Corporate Research; obrázky prebraté z článku *Visual Marker Detection and Decoding in AR Systems: A Comparative Study* [14]



Obrázok 2.3: Vnorený marker, prebraté z článku *A nested marker for augmented reality* [7]

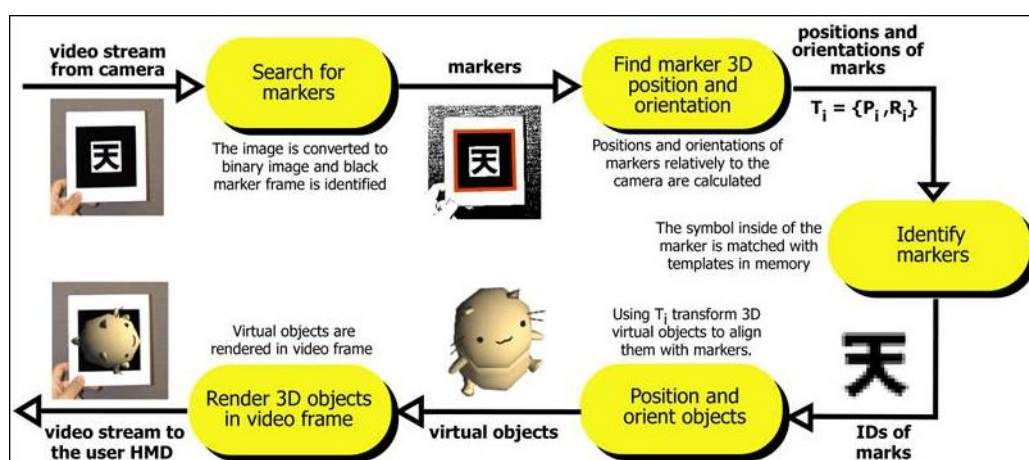
Druhým typom techník využívajúcich počítačové videnie sú techniky založené na modeli. Ich princípom je explicitné použitie modelu príznakov (napríklad CAD model) objektov ktoré sledujú a na základe ktorých počítajú pozíciu kamery. Tento prístup môže zvýšiť robustnosť a výkon aplikácie rozšírenej reality.

Poslednou kategóriou sú hybridné techniky kombinujúce použitie senzorov a počítačové videnie. Presadilo sa najmä použitie inerciálnych senzorov. Techniky počítačového videnia majú tú nevýhodu, že sú pomerne pomalé a prudké pohyby kamery môžu mať za následok

zlyhanie sledovania pozície pozorovateľa. Zotavenie sa z takéhoto zlyhania je časovo náročné a môže spôsobiť dočasné stratenie schopnosti pracovať v reálnom čase. Na rozdiel od toho, inerciálne metódy sú rýchle, robustné a dokážu odhadnúť pozíciu pozorovateľa aj v prípade rýchleho pohybu. Sú však náchylné na akumuláciu šumu.

2.3 Projekty zamerané na rozšírenú realitu

ARToolKit je framework na tvorbu aplikácií rozšírenej reality, prvýkrát predstavený na konferencii *International Workshop of Augmented Reality* v roku 1999. Je dostupný ako open-source pod GPL licenciou pre nekomerčné použitie. Framework je multiplatformný, dostupný pre Windows, Linux, Mac OS a SGI IRIX, za programovací jazyk bolo zvolené C. Na výpočet pozície kamery sa používajú štvorcové čiernobiely markery, ktoré si môže užívateľ sám nadefinovať [9]. Postup výpočtu je približený na obrázku 2.4. Farebná snímka z kamery



Obrázok 2.4: Schéma fungovania *ARToolKit-u*, prebraté z domovských internetových stránok projektu [8]

sa pomocou jednoduchého prahovania prevedie na čiernobiely, následne prebieha detekcia markerov, výpočet polohy ich vrcholov a transformačnej matice. V ďalšom kroku sa zistí, či sa jedná o ten typ markeru, ktorý je očakávaný. Pomocou transformačnej matice sa vypočíta poloha a orientácia virtuálneho objektu, ktorý sa následne vykreslí do videa z reálnej scény.

Boli vykonané testy *ARToolKit-u*, ktoré sa zamerali na subjektívne posúdenie úspešnosti. Z testov vyplynulo, že framework funguje problematicky v celom rade situácií. Neúspech spôsobovalo veľké perspektívne skreslenie, odlesky, čiastočné prekrytie markeru či jeho prehnutie. V niektorých prípadoch bol dokonca detekovaný neexistujúci marker.

Ďalším frameworkom pre tvorbu aplikácií rozšírenej reality je *Vuforia* od firmy *Qualcomm*. Tento framework je na rozdiel od *ARToolKit-u* orientovaný na mobilné platformy, menovite Android a iOS. Vývoj aplikácií uľahčuje jednoduchá integrácia frameworku do IDE *Eclipse*, *Xcode* alebo herného enginu *Unity*. V prípade použitia posledne menovaného je možné vyvíjať aplikácie súbežne pre obidve cieľové platformy. *Vuforia* umožňuje viacero spôsobov výpočtu pozície kamery. Je možné použiť obdobu klasických markerov — tzv. frame markery, alebo tzv. targety, čo sú vlastne ľubovoľné obrázky alebo fotografie. Zaujímavou možnosťou je použitie jednoduchých 3D objektov. *Vuforia* ponúka viaceré pokročilé

funkcie nad rámec samotnej rozšírenej reality, ako sú virtuálne tlačidlá reagujúce na dotyk, podpora vkladania videosekvencie do reálnej scény či využitie cloud služieb [13].

Vuforia podávala v testoch subjektívne lepšie výsledky ako *ARToolKit*. Algoritmus bol testovaný s použitím základného targetu. Problémy nespôsobovalo vysoké perspektívne skreslenie, prehýbanie targetu, odlesky ani takmer úplné prekrytie iným objektom. Detekcia a vykresľovanie boli plynulé aj na starších mobilných zariadeniach. Spomalenie a zníženie úspešnosti bolo zaznamenané akurát za znížených svetelných podmienok.

Rozšírená realita môže byť integrovaná do predmetov dennej potreby. Jednou z možností je jej uplatnenie v oblasti tzv. *Wearable Computing-u*, napríklad ako okuliare s rozšírenou realitou. Priekopníkom tejto myšlienky je Steve Mann, profesor z *University of Toronto*, ktorý zakladal projekt s názvom *EyeTap*. Cieľom tohto projektu je vytvoriť zariadenie umiestnené presne pred okom užívateľa, ktoré by mu pomocou vstavaného displeja zobrazovalo užitočné informácie [10].

Podobným projektom je *Google Glass*, ktorý však zatiaľ funguje iba ako jednoduchá kamera. Do budúcnosti sa však počíta s implementáciou prvkov rozšírenej reality. Podľa Steve Manna je veľkou nevýhodou tohto projektu to, že displej nezaberá celé zorné pole oka. Tým pádom by mohol byť pre svojho užívateľa veľmi rozptyľujúci [11].

Kapitola 3

Uniform Marker Fields

Táto kapitola sa zaoberá uniformnými poľami markerov, popisuje ich hlavnú ideu, spôsob generovania a detekcie. Tiež predkladá experimentálne výsledky a ponúka možné rozšírenia. Pokiaľ nie je uvedené inak, informácie v podkapitolách 3.1, 3.2 a 3.3 boli čerpané z článku *Uniform Marker Fields* [6].

3.1 Základná myšlienka Uniform Marker Fields

V prípade že sa pri použití klasických markerov vyžaduje voľný pohyb kamery po rozľahlejšej scéne, je potrebné ich v scéne rozmiestniť viacero. Potom je nutné v každej snímke detekovať minimálne jeden z nich. To kladie na veľkosť markerov protichodné požiadavky. Na jednej strane sa so znižujúcimi rozmermi markeru zvyšuje pravdepodobnosť, že sa vojde do záberu kamery, na druhej strane so zvyšujúcimi sa rozmermi rastie pravdepodobnosť jeho správneho detekovania. Jedným z možných riešení sú už spomínané vnorené markery pracujúce na princípe rekurzívnej štruktúry vrstiev markerov.

Ďalšiu možnosť ako implementovať rozšírenú realitu fungujúcu na rozsiahlej scéne ponúka článok *Uniform Marker Fields*. Článok predstavuje pojem *Marker Field* (ďalej pole markerov) — rovinnú štruktúru pozostávajúcu z množstva vzájomne sa prekrývajúcich častí, ktoré môžu byť chápané ako jednotlivé markery. Tieto polia sú uniformné v tom zmysle, že jednotlivé príznaky využívané na detekciu a lokalizáciu markerov sú uniformne rozložené po celej ich ploche. Pre lepšiu predstavu je pole markerov zobrazené na obrázku 3.1.

Pole markerov je založené na *Orientable Window Array* (z dôvodu neexistencie slovenského ekvivalentu nebude tento pojem preložený). Neperiodické (m, n) -window array je k -árne dvojrozmerné pole o veľkosti $h \times w$

$$A = (a_{i,j} \in \{0, \dots, k-1\}; 0 \leq i < h; 0 \leq j < w) \quad (3.1)$$

pričom platí, že sa v ňom každé okno (analógia k markeru) $A_{r,c}$ o veľkosti $m \times n$ vyskytuje práve jedenkrát. V prípade, že sú použité všetky možné okná $A_{r,c}$, (m, n) -window array sa nazýva dokonalou mapou. Dokonalá mapa tvorená binárnym neperiodickým $(2,2)$ -window array je zobrazená pomocou matice 3.2, ktorá je prebratá z publikácie *Coding Schemes for Two-Dimensional Position Sensing* [2].

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (3.2)$$



Obrázok 3.1: Pole markerov vygenerované autormi článku *Uniform Marker Fields* [6]

Takto definované *window array* však ešte nie je dostatočné pre potreby rozšírenej reality. Definícia totiž nevylučuje možnosť existencie dvoch okien (markerov), z ktorých jedno vznikne rotáciou druhého. V takomto prípade by po detekovaní jedného z týchto okien nebolo jasné o ktoré sa jedná. Tento problém rieši vlastnosť *orientovateľnosť* (anglicky *orientability*), *window arrays* potom môžu byť:

- *1-orientovateľné*: *window array* definované vyššie
- *2-orientovateľné*: *window array* u ktorého je zabezpečená *orientovateľnosť* v dvoch smeroch, čo znamená, že jednotlivé okná sú jedinečné vzhľadom k rotácii o 180°
- *4-orientovateľné*: *window array* u ktorého je zabezpečená *orientovateľnosť* v štyroch smeroch

Článok *Uniform Marker Fields* sa zaoberá binárnymi *4-orientovateľnými* n^2 -*window array*, čiže *window array* ktoré môžu byť vizualizované ako čiernobiele polia, majú štvorcový tvar okna a každé okno je jedinečné vzhľadom k rotáciám o 90° .

3.2 Generovanie poľa markerov

Podľa článku *Coding Schemes for Two-Dimensional Position Sensing* [2] sú pri použití *4-orientovateľných neperiodických* 3^2 -*window array* maximálne rozmery výsledného štvorcového poľa 12×12 . Pri takto malých rozmeroch neprináša použitie poľa markerov zásadné výhody oproti použitiu klasických markerov. Navyše, pri generovaní poľa markerov týchto rozmerov je z dôvodu zabezpečenia jedinečnosti jednotlivých markerov potrebné preskúmať stavový priestor o veľkosti 2^{144} . Pri použití 4^2 -*window array* sú maximálne rozmery 127×127 , stavový priestor pri tomto konkrétnom rozmere však narastá na 2^{16129} . V prípade 5^2 -*window array* sa dajú polia markerov rozumných rozmerov v niektorých prípadoch generovať náhodne, avšak veľkosť markeru a s tým aj plocha ktorá musí byť zachytená kamerou sa tým zbytočne zvyšuje. Z týchto dôvodov sa autori článku *Uniform Marker Fields* rozhodli použiť markery o rozmeroch 4×4 .

V prípade generovania náhodných *2-orientovateľných* alebo *4-orientovateľných* polí markerov so štvorcovými oknami (markermi) sa pravdepodobnosť existencie konfliktných okien dá vyjadriť ako 3.3:

$$p(k, h, w, n) \approx 1 - \exp\left(-\frac{o(hw)^2}{2k^{n^2} - 2k^{\lfloor \frac{n^2+1}{2} \rfloor}}\right) \quad (3.3)$$

kde k je arita, h je výška poľa markerov, w je šírka poľa markerov, n je dĺžka hrany (štvorcového) markeru a o je orientovateľnosť (2 alebo 4). Do tohto vzorca však nie sú zahrnuté prípady kedy je marker konfliktný sám so sebou (marker je rovnaký pri dvoch alebo všetkých štyroch orientáciách). Pravdepodobnosť, že v náhodne vygenerovanom poli markerov sa nachádzajú markery konfliktné samé so sebou je popísaná vzorcom 3.4,

$$p(k, h, w, n) = \left(1 - k^{\lfloor \frac{n^2+1}{2} \rfloor - n^2}\right)^{hw} \quad (3.4)$$

u ktorého je význam jednotlivých symbolov rovnaký ako u predchádzajúceho vzorca.

Výsledky experimentov s generovaním náhodných polí markerov dopadli nasledovne. Pri rozmeroch poľa 15×15 sa v niektorých prípadoch podarilo splniť vlastnosť *4-orientovateľného array window* bez dodatočných úprav výsledného poľa. V prípade rozmerov 50×50 boli konflikty nevyhnutné, no po určitom množstve náhodných zmien konfliktných markerov sa ich podarilo všetky odstrániť. Pri vyšších rozmeroch poľa markerov, napríklad 90×90 , je už množstvo potrebných zmien natoľko veľké, že celý proces sa stáva veľmi zdĺhavým.

Z tohto dôvodu sa autori článku *Uniform Maker Fields* rozhodli využiť na generovanie polí markerov generický algoritmus. Prvotná populácia je kópiami toho istého poľa markerov, prípadne môže byť každý jedinec generovaný náhodne. Hodnotiaca funkcia je definovaná vzorcom 3.5,

$$f(A) = \frac{1}{c(A) + 1} \quad (3.5)$$

kde $c(A)$ vyjadruje počet konfliktných markerov v rámci poľa markerov A , čiže hodnota hodnotiacej funkcie je nepriamo úmerná k počtu konfliktných markerov. Iterovanie algoritmu končí, keď hodnotiaca funkcia dosiahne hodnotu 1, čo znamená, že bolo nájdené pole bez konfliktných markerov. Mutácia je definovaná ako nahradenie jedného markeru iným, náhodne vygenerovaným. Výber nahradzovaného markeru je náhodný, konfliktné markery majú však väčšiu pravdepodobnosť, že budú vybraté. Z dôvodu zrýchlenia konverencie sa po vygenerovaní náhodného nahradzujúceho markeru prevedú kontroly, či sa taký marker v poli markerov už nenachádza, prípadne či ešte nezvýši počet konfliktov. Ak áno, takýto marker nie je použitý.

Nakoľko aj takýto prístup je výpočtovo veľmi náročný, autori sa rozhodli pre distribuovaný výpočet na klient-server architektúre. Úlohou serveru je uchovávať aktuálnu populáciu a rozdeľovať úlohy jednotlivým klientom. Nakoľko doba výpočtu u klientov sa môže diametrálne odlišovať, server nepracuje s rôznymi generáciami jedincov ale udržiava si jedinu populáciu, ktorej jedincov postupne aktualizuje. Klientovi je zaslaný jediný jedinec, ktorý podstúpi proces mutácie. Pokiaľ je výsledkom zmenený jedinec, ktorý má rovnaký alebo nižší počet konfliktov, je tento jedinec zaslaný späť na server. Maximálne rozmery polí markerov ktoré sa podarilo takýmto spôsobom vygenerovať sú zobrazené v tabuľke 3.1.

pomer strán	maximálne rozmery
1 : 1	92 × 92
$\sqrt{2}$: 1	110 × 78
2 : 1	122 × 61
3 : 1	159 × 53

Tabuľka 3.1: Maximálne rozmery polí markerov pri rozličných pomeroch strán

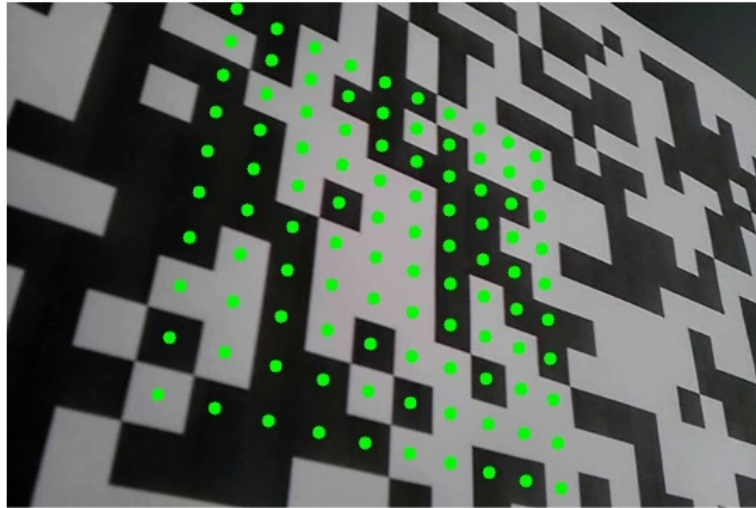
3.3 Detekcia markerov a experimentálne výsledky

Prvou fázou detekcie je extrakcia úsečiek tvorených hranami jednotlivých markerov, tzv. *edgelov*. Postupuje sa tak, že sa na obraz aplikuje mriežka horizontálnych a vertikálnych skenovacích čiar. Približne 13 čiar v každom smere je dostatočných na správne fungovanie algoritmu. Následne sa na týchto čiarach detekujú hranové body, ktoré odpovedajú prechodu medzi bielou a čiernou farbou. To prebieha pomocou *adaptívneho prahovania* nad pohyblivým priemerovacím oknom. Následne sa použije *Sobelov operátor* na výpočet gradientu, ktorý je kolmý na hranu prechodu (*edgel*). Z dôvodu zvýšenia presnosti detekcie sa v oboch smeroch nájdeného *edgelu* hľadajú ďalšie vrcholy na hranách a to podľa vzorca 3.6:

$$\tilde{p}_{i+1} = p_0 + iw\tilde{s}_i \quad (3.6)$$

kde \tilde{p}_{i+1} pre $i \in \{1, 2, \dots\} \cup \{-1, -2, \dots\}$ je hľadaný vrchol, p_0 počiatočný vrchol, w je vzdialenosť posunutia v smere pôvodného *edgelu* a \tilde{s}_i je normalizovaný smerový vektor pôvodného *edgelu*. Ak je takýto vrchol nájdený, rovnica pôvodného *edgelu* sa patrične upraví. Do ďalších fáz algoritmu sa dostanú iba tie *edgely*, ktoré sú dostatočne spoľahlivé, teda bol pre ne nájdený dostatočný počet bodov p_i a tie sú kolineárne v rámci určitej tolerancie. Ďalším krokom je vypočítanie dvoch dominantných úbežných bodov. Na výpočet každého z nich sa použijú priamky preložené cez *edgely*. Priamka prechádzajúca cez úbežné body tvorí horizont. V ďalšom kroku sa tieto dva body použijú na výpočet mriežky, reprezentujúcej hranice medzi jednotlivými dlaždicami v poli markerov. Táto mriežka je tvorená priamkami prechádzajúcimi práve cez jeden z úbežných bodov. Nakoniec sa s použitím mriežky extrahujú hodnoty dlaždíc poľa markerov. Konečný výsledok je zobrazený na obrázku 3.2.

Testovanie prebiehalo na databáze 36 videí s týmito charakteristickými akciami: zoom kamery, horizontálny pohyb nad poľom markerov, rotácia, perspektívne skreslenie, obecný pohyb a prekrytie časti poľa inými objektmi. Videá boli natáčané tromi typmi mobilných telefónov, testy prebiehali nad dvomi poľami s rozdielnou hustotou markerov. Výsledky ukázali, že algoritmus je schopný správne pracovať aj v náročných scénach zahŕňajúcich rozmazanie spôsobené rýchlym pohybom, rotáciu či silné perspektívne skreslenie. Najvyššiu úspešnosť, vyššiu ako 99%, sa podarilo dosiahnuť pri horizontálnom pohybe, rotácii a perspektívnom skreslení. Úspešnosť ostatných výsledkov neklesla pod 90%. Pole s nižšou hustotou markerov vykazovalo väčšiu úspešnosť, čo bolo spôsobené dlhšími hranami markerov a teda presnejšími výpočtami *edgelov*. Výnimkou bola scéna s prekrývaním, kedy väčšie množstvo informácie obsiahnutej v hustejšom poli markerov malo za následok vyššiu úspešnosť. Percento navštívených pixlov vstupného obrazu sa držalo na nízkej úrovni, hodnote menšej ako 6%.



Obrázok 3.2: Extrakcia dlaždíc poľa markerov, ich stredy sú reprezentované zelenými bodmi, obrázok prebratý z domovskej stránky polí markerov [12]

3.4 Rozšírenia Uniform Marker Fields

Táto podkapitola popisuje možné rozšírenia, o ktoré by sa dali vylepšiť myšlienky z článku *Uniform Marker Fields*. Implementácia týchto rozšírení bude predstavovať gro praktickej časti tejto práce.

V prvom rade by mohlo dôjsť k zvýšeniu počtu použitých farieb v poli markerov. To by malo mať za následok zníženie počtu konfliktných markerov v porovnaní s pôvodným prístupom. Od toho sa odvíja aj možnosť vygenerovať polia markerov omnoho väčších rozmerov alebo použitie markerov s menšími rozmermi. Algoritmus generovania by sa mohol dať upraviť tak, aby sa jednotlivé markery zafarbovali systematicky za účelom vytvorenia farebných obrazcov. Možnou cestou by mohlo byť načítanie obrazovej predlohy a vygenerovanie poľa markerov na jej základe. V prípade konfliktov by sa na ich odstránenie mohli využiť techniky podobné tým, ktoré boli popísané v článku *Uniform Marker Fields*, alebo by mohli byť vyvinuté nové, ktoré by brali ohľad na zachovanie určitej podobnosti s predlohou. Vďaka možnosti generovať polia farebných markerov na základe predlohy a vo veľkých rozmeroch by mohol tento prístup nájsť uplatnenie pri tvorbe hier s rozšírenou realitou. Vývojár by bol schopný na základe ním nadefinovanej mapy herného prostredia vytvoriť pole markerov, ktoré by mu umožnilo dokresliť do reálnej scény rozsiahly virtuálny svet a zároveň by samo, z určitej vzdialenosti, pripomínalo mapu herného prostredia.

Problémom pri použití viacerých farieb by mohla byť detekcia za rôznych svetelných podmienok a pri použití rôznych kamier. Preto by bolo potrebné vymyslieť alternatívny spôsob reprezentácie markerov.

Kapitola 4

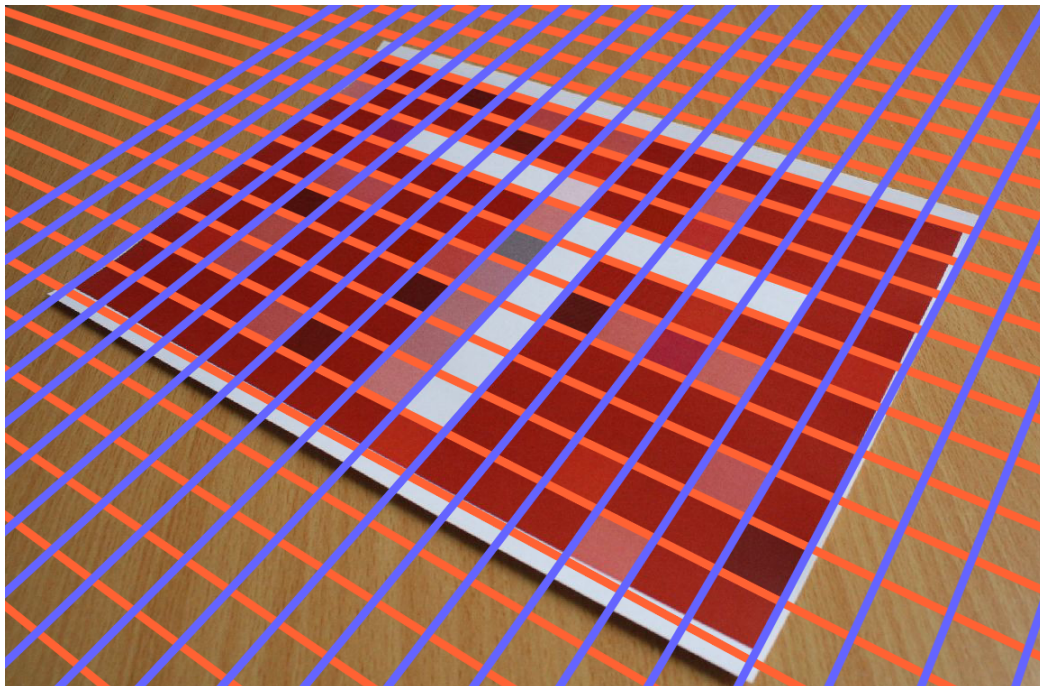
Implementácia algoritmov na generovanie a detekciu polí markerov

Jedným z cieľov praktickej časti tejto diplomovej práce je realizovať rozšírenia článku *Uniform Marker Fields* definovaných v predchádzajúcom texte. Táto kapitola popisuje vývoj aplikácie na generovanie polí markerov ako aj nástroja na ich detekciu a vyhodnocovanie úspešnosti. Z dôvodu že táto práca je viac výskumného ako implementačného charakteru, sú v tejto kapitole popísané iba obecné postupy a rozhodnutia, ktoré boli učené počas vývoja. Implementačné detaily sa nachádzajú v dokumentácii k zdrojovým kódom v adresári `\doc\doxygen` na priloženom DVD.

4.1 Použité technológie

Pre potreby tejto diplomovej práce boli vytvorené tri programy. Aplikácia na generovanie polí markerov (*MFGGenerator*), program na testovanie úspešnosti detekcie prechodov (*transitionDetector*) a nástroj na štúdiu stálosti farieb (*colourTester*), ktorý je popísaný v ďalšej kapitole. Všetky tri programy boli implementované ako konzolové, bez grafického užívateľského rozhrania. Ich ovládanie prebieha pomocou parametrov zadávaných cez príkazový riadok. Možnosti ovládania sú zhrnuté v prílohe B. Ako programovací jazyk bolo zvolené C++, implementácia prebiehala vo vývojovom prostredí *Microsoft Visual Studio 2010*. Všetky zdrojové texty spolu s projektovými súbormi `.sln` sa nachádzajú v adresári `\src` na priloženom disku DVD. Počas vývoja bol kladený dôraz na podrobné komentovanie zdrojových súborov, dokumentácia vygenerovaná programom *Doxygen* sa nachádza v adresári `\doc\doxygen`. Na zjednodušenie práce s obrazom bola použitá knižnica *OpenCV* vo verzii 2.4.2. Pre prezentačné účely bol vytvorený plagát a krátke video.

Práca tiež využíva program na detekciu mriežky v obrázku obsahujúcom pole markerov. Tento program bol v binárnej forme poskytnutý Ústavom počítačovej grafiky a multimédií na FIT VUT. Program je schopný detekovať mriežku v ľubovoľnom dvojrozmernom poli štvorcových políčok, pre potreby tejto práce je vstupom fotografia vytlačeného poľa markerov. Vizuálnym výstupom je vstupný obrázok s detekovanou mriežkou, ukážka výstupu je zobrazená na obrázku 4.1. Rozmery mriežky nemusia byť totožné s rozmermi poľa markerov. Textový výstup obsahuje súradnice stredov políčok v detekovanej mriežke. Program sa nachádza v adresári `\bin\gridDetector`, ovládanie je popísané v prílohe B.



Obrázok 4.1: Detekovaná mriežka vo fotografii poľa markerov

4.2 Program na generovanie polí markerov

Účelom programu je implementovať všetky rozšírenia práce Uniform Marker Fields definované v podkapitole 3.4. Vstupom je ľubovoľný farebný obrázok v pôvodnej veľkosti, ktorý je po načítaní podvzorkovaný na želané rozmery výsledného poľa markerov. Rozsah použitých farieb nie je nijakým spôsobom obmedzený. Následne je tento obrázok pomocou genetického algoritmu upravovaný tak, aby čo najlepšie spĺňal tieto tri obmedzenia:

- minimálne množstvo konfliktov markerov
- maximálna podobnosť so zdrojovým obrázkom
- maximálna úspešnosť pri detekcii markerov

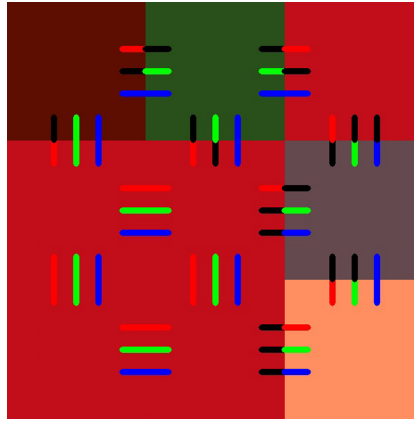
4.2.1 Reprezentácia markeru

V prístupe z článku *Uniform Marker Fields* bol marker definovaný priamo farbami jednotlivých dlaždíc, z ktorých sa skladal. Nakoľko bol počet farieb obmedzený na dve, čiernu a bielu, bolo bezproblémové ich vo fáze detekcie rozlíšiť. Takýto prístup je však nepoužiteľný pre potreby tejto práce, ktorá si kladie za cieľ použiť 24 bitovú farebnú hĺbku. Pri takomto počte farieb by sa len vo výnimočných prípadoch podarilo detekovať rovnakú farbu ako bola vygenerovaná. Je to spôsobené nedokonalosťou zariadení na tlač a snímanie, či pôsobením rozdielnych svetelných podmienok. Z tohto dôvodu bola navrhnutá alternatívna reprezentácia markerov. Namiesto konkrétnych farieb bude marker definovaný farebnými zmenami medzi susediacimi dlaždicami a to pre každý kanál z RGB farebného modelu zvlášť. Pre každú vnútornú hranu markeru medzi dvomi dlaždicami je treba určiť, či sa pri prechode z prvej dlaždice na druhú intenzita v červenej, zelenej a modrej zložke zvyšuje,

znižuje, alebo zostáva približne rovnaká. Posledný prípad nastáva ak je absolútna hodnota rozdielu intenzít menšia ako 30. Táto hodnota bola určená empiricky.

Jeden marker s rozmermi 3×3 je teda definovaný tridsiatimi šiestimi prechodmi intenzít, pričom každý z nich môže nadobudnúť tri hodnoty. Situácia je znázornená na obrázku 4.2. Úsečka konštantnej farby značí, že intenzity dlaždíc sú v danom farebnom kanáli približne rovnaké, čierna farba na jednom konci úsečky označuje dlaždicu s nižšou intenzitou v danom kanáli. Z implementačného hľadiska je marker reprezentovaný poľom tridsiatich šiestich znakov, ktoré nadobúdajú jednu z troch hodnôt podľa typu prechodu, ktorému prináležia.

Tento prístup je založený na nasledovnej hypotéze. Pokiaľ bude pole markerov počas tlače alebo snímania celoplošne ovplyvnené nejakým faktorom (napr. osvetlené farebným svetlom), tak bude zmena všetkých farieb približne konštantná. To by malo mať za následok minimálne zmeny v prechodoch medzi jednotlivými dlaždicami. V prípade lokálnych faktorov, ako napr. tieň vrhnutý na časť poľa markerov, by vyššie uvedená hypotéza mala platiť pre každú konštantne ovplyvnenú časť poľa markerov.



Obrázok 4.2: Marker s rozmermi 3×3 je definovaný tridsiatimi šiestimi prechodmi intenzít

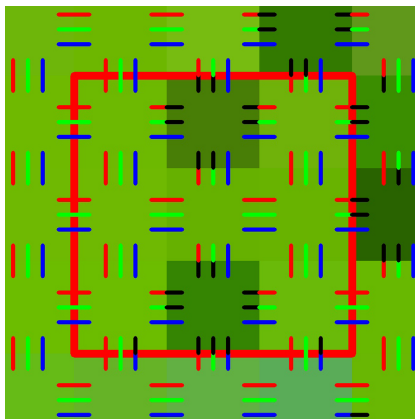
Už v začiatkoch implementácie tohto programu bolo rozhodnuté, že markery budú mať rozmery 3×3 . Zmenšenie veľkosti z 4×4 prezentovanej článkom *Uniform Marker Fields* umožnila rozdielna reprezentácia markeru, ktorá poskytuje až 3^{36} ich rôznych kombinácií (bez odpočítania markerov konfliktných so sebou samými). Výhodou zmenšenia je, že pri snímaní poľa markerov kamerou postačuje nasnímať menšiu plochu na to, aby bolo možné vypočítať polohu kamery. Naopak, nevýhodou je väčšia pravdepodobnosť výskytu konfliktov. Testy však preukázali, že aj pri použití markerov s veľkosťou 4×4 sa v poli markerov vytvorenom priamym podvzorkovaním vstupného obrázka nachádzali konflikty, ktoré bolo následne treba odstrániť. Ďalšie zmenšenie rozmerov na 2×2 bolo zavrhnuté, keďže by to malo za následok zníženie počtu použiteľných markerov na 3^{12} (bez odpočítania markerov konfliktných so sebou samými).

4.2.2 Vyhladávanie konfliktov

Pri vyhľadávaní konfliktov je potrebné porovnať každý marker s každým. Počet porovnaní pre pole markerov s rozmermi $m \times n$ je vyjadrený vzorcom 4.1,

$$\sum_{x=1}^{(m-2)(n-2)-1} x \quad (4.1)$$

kde $(m - 2)(n - 2)$ je počet markerov o veľkosti 3×3 obsiahnutých v poli. Pre ilustráciu, v prípade poľa markerov o veľkosti 100×100 je počet potrebných porovnaní väčší ako 44 miliónov. Navyše, ako bude popísané v ďalšej podkapitole, pole markerov sa upravuje iteratívne, takže vyhľadávanie konfliktov je počas behu programu vykonávané opakovane. Výpočet ďalej komplikuje požiadavka *4-orientovateľnosti*, čiže v rámci poľa markerov nemôžu existovať dva markery, z ktorých jeden je možné vytvoriť rotáciou toho druhého. Z týchto dôvodov bol kladený veľký dôraz na efektívnosť algoritmu na vyhľadávanie konfliktov.



Obrázok 4.3: Marker konfliktný sám so sebou

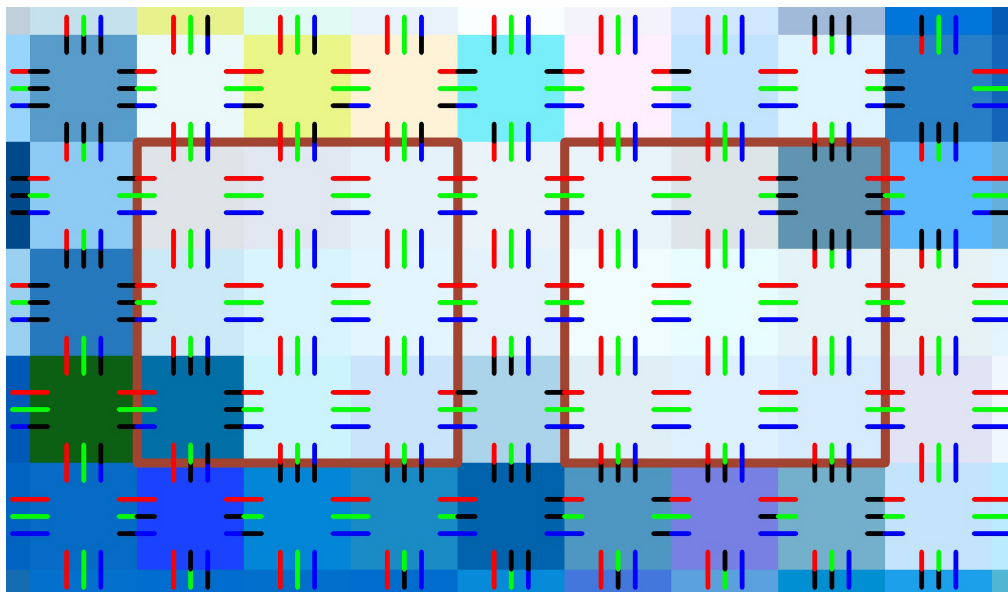
Ako bolo spomenuté v predchádzajúcej podkapitole, marker je reprezentovaný poľom znakov, tzv. deskriptorom. Pre každý marker sa vypočítajú štyri deskriptory, reprezentujúce každú z jeho možných rotácií. V tomto bode sa porovnaním deskriptorov zisťuje konflikt markeru so sebou samým. Následne je vybratý deskriptor s najmenšou hodnotou (po prevedení poľa znakov na číslo) a tento je použitý pre porovnávanie s ostatnými markermi. Vďaka tomuto prístupu nie je nutné porovnávať pri dvojici markerov všetky kombinácie ich možných rotácií, je to implicitne zaručené porovnaním ich deskriptorov s najnižšou hodnotou.

Obrázok 4.3 zobrazuje marker konfliktný sám so sebou. Na obrázku 4.4 je fialovou farbou zvýraznená dvojica konfliktných markerov

4.2.3 Genetický algoritmus

Pri testovaní algoritmu na vyhľadávanie kolízií bolo zistené, že ku konfliktom dochádza už pri malých rozmeroch výsledného poľa markerov a to aj v prípade, že sa v zdrojovom obrázku nenachádzajú veľké plochy konštantnej farby. Prvotným riešením bola náhodná zmena farieb dlaždíc u konfliktných markerov. Tento prístup síce viedol k odstráneniu konfliktov, avšak zvyšné dve stanovené kritéria, maximálna podobnosť so zdrojovým obrázkom a maximálna úspešnosť pri detekcii markerov, sa dali ovplyvňovať len nepriamo. Preto bolo rozhodnuté použiť genetický algoritmus, v ktorom jedinec bude reprezentovaný celým poľom markerov a všetky požiadavky na jeho kvalitu sa zahrnú do hodnotiacej funkcie.

Idea fungovania programu na generovanie poľa markerov s využitím genetického algoritmu je nasledovná. Po načítaní vstupného obrázku a jeho podvzorkovaní na želané rozmery sa týmto prvotným poľom markerov inicializuje nultá populácia. Predvoleným spôsobom je vytvorenie rovnakých jedincov duplikáciou prvotného poľa markerov, voliteľne je možné



Obrázok 4.4: Konfliktné markery

aplikovať náhodný šum. Počas každej evolúcie sa na jedincov aplikujú metódy reprodukcie a mutácie, kríženie nie je použité. Jedince, čiže polia markerov, sú implementované pomocou obrázka, takže mutácia pozostáva zo zmeny jeho farieb. Problematika mutácie je podrobne rozobraná neskôr v tejto podkapitole. Všetky jedince sú následne ohodnotené hodnotiacou funkciou, do ďalšej evolúcie prežívajú len tie najlepšie. Genetický algoritmus končí po želanom počte evolúcií, výstupom programu je najlepší jedinec.

Najväčší dôraz pri implementácii genetického algoritmu bol kladený na hodnotiacu funkciu. Práve od jej správnosti závisí, či bude program schopný generovať použiteľné polia markerov. Výsledkom tejto funkcie nie je iba jedno číslo hodnotiace zdatnosť celého jedinca. Pre potreby mutačného algoritmu priradí táto funkcia zdatnosť každej dlaždici na základe toho, ako dobre spĺňa každú z troch podmienok na kvalitu poľa markerov uvedených vyššie v tejto kapitole. Alternatívne sa dá hodnotiacia funkcia chápať ako súčet troch čiastkových funkcií. Navyše, príspevky týchto funkcií sú vážené, takže užívateľ môže nastavením váh ovplyvňovať charakter výsledného poľa markerov. Celková zdatnosť jedinca sa potom vypočíta ako priemerná zdatnosť jeho dlaždíc.

Prvá čiastková funkcia slúži v konečnom dôsledku na odstránenie konfliktných markerov. V každej evolúcii sa pre každého jedinca vypočítajú prechody medzi susednými dlaždicami, ktoré sú následne využité na vyhľadanie konfliktných markerov. Táto čiastková funkcia priradzuje hodnotenie každej dlaždici na základe toho, či je súčasťou nejakého konfliktného markeru alebo nie. Implementačne je to riešené tak, že sa prechádza zoznam konfliktných markerov a každej príslušnej dlaždici sa zníži zdatnosť o konštantnú hodnotu. Výhodou tohto prístupu je, že zdatnosť dlaždíc prináležiacich viacerým konfliktným markerom bude znížená niekoľkonásobne. To má za následok, že počas mutácie sa zvýši pravdepodobnosť zmeny takejto dlaždice, čo môže priniesť odstránenie viacerých kolízií naraz.

Druhá čiastková funkcia zaručuje čo najväčšiu podobnosť výsledného poľa markerov s pôvodným obrázkom. Každá dlaždica aktuálne spracovávaného jedinca sa porovná s príslušnou dlaždicou prvotného poľa markerov, ktoré vzniklo podvzorkovaním vstupného obrázka.

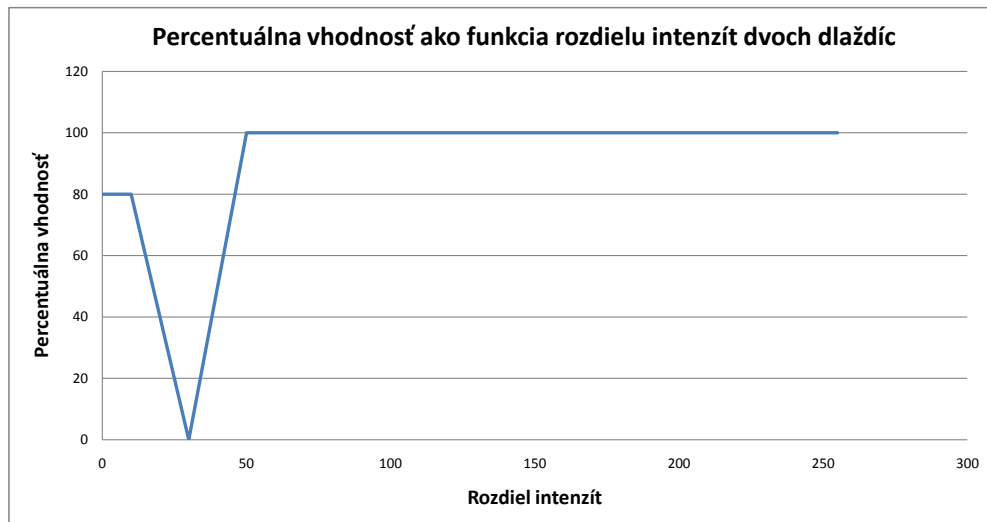
Zdatnosť dlaždice n popisuje rovnica 4.2,

$$f[n] = 255 - \frac{\sum_{ch=1}^3 |o[n]_{ch} - a[n]_{ch}|}{3} \quad (4.2)$$

v ktorej 255 je maximálny rozdiel intenzít v jednom kanáli pri 8 bitoch na kanál, 3 je počet použitých kanálov, $o[n]_{ch}$ je intenzita n -tej dlaždice z pôvodného poľa markerov v kanáli ch a $a[n]_{ch}$ je intenzita n -tej dlaždice z aktuálne spracovávaného poľa dlaždíc v kanáli ch . Ide o výpočet priemernej zmeny intenzity na kanál a jeho odpočítanie od maximálnej možnej priemernej zmeny. Funkčné hodnoty sú v intervale $\langle 0, 255 \rangle$, vyššia hodnota značí väčšiu zdatnosť dlaždice.

Druhou možnosťou je porovnávanie vygenerovaných farieb s predlohou pomocou farebného modelu YUV s uprednostňovaním zmien v zložke Y. Cieľom je, aby sa pole markerov od zdrojového obrázka líšilo viac v luminancii ako v chrominancii farieb, čo by mohlo pôsobiť subjektívne menej rušivo.

Úlohou poslednej čiastkovej funkcie je zvýšiť úspešnosť detekcie prechodov medzi dlaždicami a tým aj úspešnosť správneho rozpoznania markerov. Ako bolo spomínané vyššie v tejto kapitole, či bude prechod intenzít medzi dvomi dlaždicami klasifikovaný ako konštantný alebo nie, sa rozhoduje na základe porovnania rozdielu intenzít s prahom, prednastaveným na 30. Prvotné testovanie ukázalo, že pokiaľ je rozdiel intenzít dvoch dlaždíc v okolí tohto prahu, pravdepodobnosť správnej detekcie typu prechodu klesá. Je to spôsobené tým, že aj malá zmena rekonštruovaných farieb z vytlačeného poľa markerov oproti farbám vygenerovaným môže spôsobiť, že namiesto konštantného prechodu bude detekovaná stúpajúca resp. klesajúca intenzita a vice versa. Bolo treba zabezpečiť, aby boli favorizované také dvojice dlaždíc, u ktorých bude rozdiel intenzít dostatočne vzdialený od použitého prahu. Druhou, menej dôležitou, požiadavkou bolo uprednostniť silné prechody medzi dlaždicami. Dôvodom bola snaha o zvýšenie presnosti detekcie mriežky vo fotografii poľa markerov. Obidve požiadavky zahŕňa funkcia na obrázku 4.5. Na vodorovnej osi sa nachádza

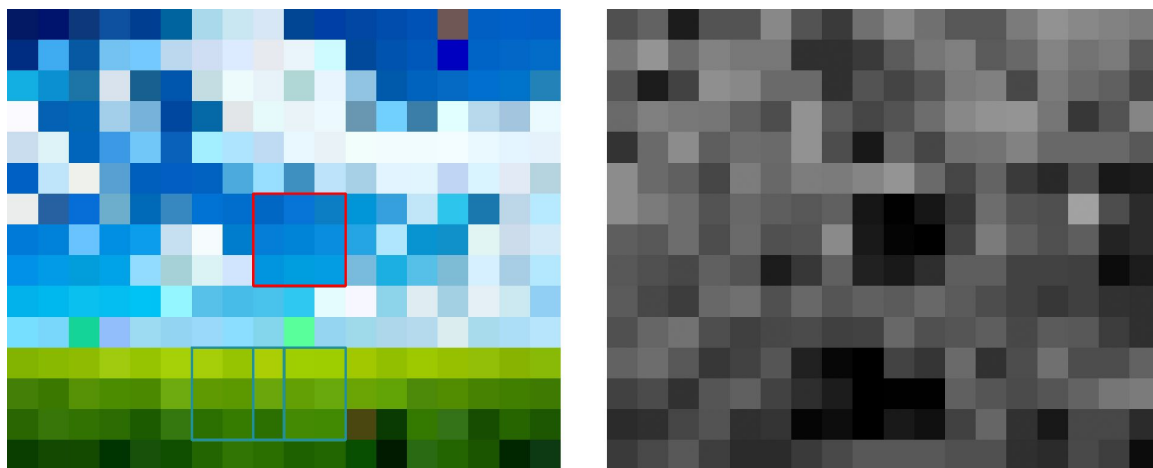


Obrázok 4.5: Čiastková funkcia na zvýšenie úspešnosti detekcie prechodov medzi dlaždicami

absolútna hodnota rozdielu intenzít dvoch dlaždíc. Zvislá os vyjadruje percentuálnu časť maximálnej zdatnosti. Na grafe si je možné všimnúť, že pri rozdiel intenzít rovnom 30 je výsledná zdatnosť nulová. Ako sa rozdiel intenzít vzdáva od tejto hodnoty, vhodnosť stúpa až na 80% resp. 100%. Tieto dve čísla sú rozdielne práve z dôvodu uprednostnenia silných prechodov. Body zlomu boli určené empiricky. Funkcia nebola navrhnutá hladšie z dôvodu, aby bolo jej vyčíslovanie výpočtovo čo najjednoduchšie. Dochádza k nemu až dvanásťkrát pre každú dlaždicu. Počíta sa totiž jej vhodnosť s každou susednou dlaždicou a to pre každý kanál zvlášť. Výsledné hodnotenie dlaždice je priemerom týchto hodnôt.

Na obrázku 4.6 sa nachádza ukážka poľa markerov vygenerovaného z fotografie krajiny a grafické zobrazenie hodnotenia, ktoré k nemu prislúcha. Pole markerov obsahuje jeden marker konfliktný sám so sebou a jednu čiastočne sa prekrývajúcu dvojicu konfliktných markerov. Hodnotenie zahŕňa výsledky všetkých troch čiastkových funkcií, ktoré boli prevedené do intervalu $\langle 0, 255 \rangle$ a zobrazené ako odtieň šedej. Svetlá farba predstavuje vysokú vhodnosť, tmavá nízku. Je vidno, že v miestach konfliktov došlo k výraznému stmaveniu, najviac evidentné je to v mieste prekrytia dvoch konfliktných markerov. Naopak, vysokú vhodnosť má napríklad modrá dlaždica v siedmom riadku od vrchu a treťom stĺpci sprava. Vysoké ohodnotenie vrátili všetky tri čiastkové funkcie. V pôvodnom obrázku na pozícii tejto dlaždice presvitala obloha pomedzi mraky, preto jej modrá farba silne koreluje so zdrojovým obrázkom. Navyše, nie je súčasťou žiadneho konfliktného markeru a zmeny intenzít medzi ňou a okolitými dlaždicami sú výrazné.

Takto vytvorené dvojrozmerné pole vhodností je využívané pri mutácii. Tá pozostáva zo zmeny farby nula až šiestich dlaždíc. Ich výber je náhodný len pre nula až tri z nich, zvyšné nula až tri dlaždice sú vybrané v poradí od najnižšej vhodnosti. Zmena farby dlaždice pozostáva z jednej až troch iterácií, v každej z nich sa vyberie jeden z kanálov RGB a jeho intenzita je náhodne zmenená. Snahou pri implementácii algoritmu mutácie bolo zabezpečiť čo najväčšiu náhodnosť, výber dlaždíc v poradí podľa vhodnosti bol použitý na zrýchlenie konvergenencie celého genetického algoritmu. Všetky vyššie uvedené čísla boli stanovené empiricky, počas prvotného testovania programu.



Obrázok 4.6: Pole markerov a jeho hodnotenie

4.3 Program na testovanie úspešnosti detekcie

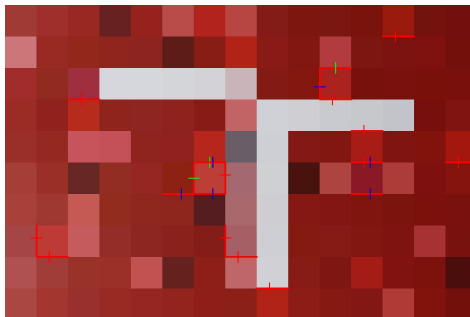
Účelom tohto programu je zistiť, aké percento prechodov a markerov sa podarilo správne detekovať po vytlačení poľa markerov a jeho nasnímaní kamerou. Nižšie v texte sú používané dva pojmy, vygenerované a rekonštruované pole markerov. Tým prvým sa myslí to pole, ktoré je výstupom programu *MFGenerator*. Druhý pojem označuje také, ktoré bolo rekonštruované z fotografie vygenerovaného poľa markerov. V ideálnom prípade by ich odpovedajúce si dlaždice mali rovnaké farby.

Vstupom programu sú dva textové súbory a fotografia poľa markerov. Prvý z textových súborov obsahuje zoznam horizontálnych a vertikálnych prechodov z vygenerovaného poľa markerov. V tom druhom sa nachádza výstup programu *gridDetector*, teda stredy dlaždíc v mriežke zdetekovanej vo fotografii poľa markerov.

Program pracuje nasledovne. Najskôr sa do dátových štruktúr programu načítajú prechody z prvého textového súboru. Z toho druhého sa importujú stredy políčok z detekovanej mriežky a preusporiadajú sa do všetkých štyroch možných rotácií. Tento krok je potrebný k správnej lokalizácii poľa markerov v rámci mriežky, keďže jeho fotografia môže byť urobená z ľubovoľného pohľadu. Následne sa pre každú zo štyroch rotácií mriežky určia typy horizontálnych a vertikálnych prechodov medzi jej políčkami. Na tento úkon je potrebné rekonštruovať farbu týchto políčok. Tá sa z dôvodu snahy o minimalizovanie nepresností spôsobených šumom určí ako priemer farieb pixlov v okolí 5×5 okolo stredu políčka. Ďalším krokom je nájdenie najlepšej zhody medzi prechodmi vygenerovaného poľa markerov a prechodmi medzi políčkami mriežky. Túto operáciu komplikuje fakt, že rozmery mriežky môžu byť väčšie ako rozmery vygenerovaného poľa markerov. Preto je nutné porovnať každú časť mriežky (a to pre každú zo štyroch rotácií) s vygenerovaným poľom markerov a nájsť takú, ktorá vykazuje najmenší počet rozdielnych typov prechodov. Tá je potom vyhlásená za rekonštruované pole markerov.

Textové výstupy programu sú dva. Prvým je percentuálne vyjadrenie úspešnosti detekcie prechodov. V prípade použitia farebných polí markerov v reálnych aplikáciách rozšírenej reality je však dôležitejší iný parameter. Tým je počet bezchybne detekovaných markerov, na základe ktorých je možné vypočítať korektnú polohu kamery. Práve táto hodnota je druhým textovým výstupom programu. Oproti úspešnosti prechodov bude spravidla nižšia, keďže jeden nesprávne detekovaný prechod môže spôsobiť chybu až u šiestich markerov. Grafický výstup programu predstavuje rekonštruované pole markerov a to v troch variantoch: bez vizualizácie prechodov, s vizualizáciou chybných alebo všetkých prechodov.

Rekonštruované pole markerov s vizualizovanými chybnými prechodmi je zobrazené na obrázku 4.7.



Obrázok 4.7: Rekonštruované pole markerov

Kapitola 5

Vyhodnotenie úspešnosti detekcie a štúdie stálosti farieb

Táto kapitola reflektuje chronologický postup v práci po implementovaní dvoch programov popísaných v predchádzajúcej kapitole. Jej prvá časť sa venuje testovaniu úspešnosti detekcie prechodov a markerov vo fotografiách vygenerovaných polí markerov. Následne sa práca zameriava na odhalenie dôvodov nízkej úspešnosti a jej zlepšenie. Ide o štúdiu vlastností farieb, konkrétne ich zmien v procese tlače a snímania kamerou. Štúdia presahuje rámec rozšírenej reality, získané poznatky môžu byť využité v aplikáciách, kde dochádza ku kódovaniu informácií pomocou farieb. Táto časť rozširuje zadanie diplomovej práce a predstavuje jej hlavný prínos. Pre potreby tejto štúdie bol vytvorený ďalší program. Z dôvodu väčšej ucelenosti a logickej nadväznosti textu je jeho implementácia popísaná v tejto kapitole, nie v predchádzajúcej. V závere je diskutované začlenenie poznatkov do programu na generovanie polí markerov a následné zvýšenie úspešnosti detekcie.

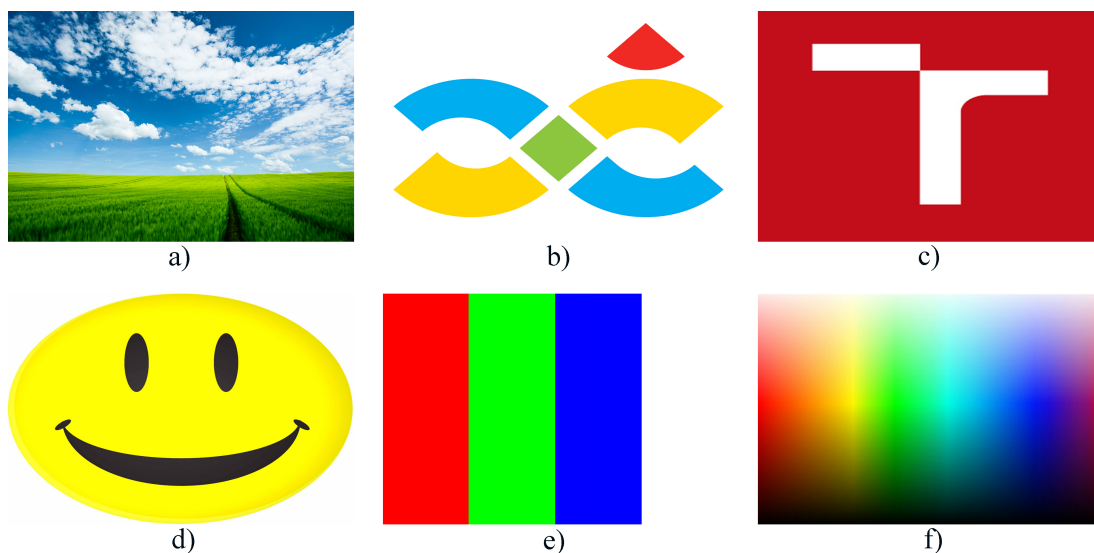
5.1 Úspešnosť detekcie prechodov a markerov

Táto podkapitola popisuje testovanie úspešnosti detekcie typov prechodov a markerov. Správne detekovaným prechodom sa rozumie taký prechod v jednom farebnom kanáli, ktorý bol rovnakého typu vo vygenerovanom ako aj v rekonštruovanom poli markerov. Správne detekovaný marker je taký, ktorý sa skladá výlučne zo správne detekovaných prechodov. Podkapitola sa nezaoberá presnosťou detekcie mriežky ani výpočtom polohy kamery.

Zdrojové fotografie a logá ktoré boli použité na generovanie polí markerov sú zobrazené na obrázku 5.1. Na ich názvy je odkazované po zvyšok tejto kapitoly. Polia markerov z nich boli vygenerované s rozmermi 15×10 a boli vytlačené na papier o veľkosti A4. Hrana jednej dlaždice mala tým pádom dĺžku približne 2cm. Pri menších rozmeroch dlaždíc by mohli u niektorých kamier nastávať problémy so zaostrovaním v prípade detailného záberu na malú časť poľa markerov. Testovanie prebiehalo na počítači s procesorom *Intel Core i5-2520M* s dvomi jadrami taktovanými na 2.5GHz, štyrmi GB operačnej pamäte a operačným systémom Windows 7 64bit. Pri tejto konfigurácii trvalo vygenerovanie poľa markerov s rozmermi 10×15 , za použitia 10 jedincov a 4000 evolúcií genetického algoritmu približne 40 minút. So zvyšujúcim sa počtom jedincov, evolúcií a hlavne so zväčšujúcimi sa rozmermi výsledného poľa markerov čas potrebný na dokončenie programu prudko narastal. Testovanie prebiehalo za použitia 10 až 20 jedincov a 1000 až 4000 evolučných cyklov. Vygenerované polia markerov boli následne vytlačené a odfotené pomocou rôznych zariadení v rozličných

svetelných podmienkach. Keďže aplikácie využívajúce rozšírenú realitu sú vo väčšine prípadov ciele na mobilné platformy, dôraz bol kladený práve na snímanie pomocou mobilných telefónov a tabletov, menovite *HTC One V*, *Samsung Galaxy S*, *Nokia Lumia 800* a *Huawei MediaPad*. Pre porovnanie bola použitá aj digitálna zrkadlovka *Canon EOS 1000D*. Fotografie vytlačených polí markerov boli spracované programom *gridDetector*. Výstup tohto programu, teda súradnice stredov polí v mriežke, spolu so zoznamom typov prechodov vo vygenerovanom poli markerov boli vstupom pre program *transistionDetector*, ktorý vypočítal úspešnosť detekcie.

Tá však bola pomerne nízka. Výsledky vybraných testov sú v tabuľke 5.1. Stĺpec s názvom w_s označuje váhu od 0 po 1, ktorá bola nastavená pre čiastkovú hodnotiacu funkciu zaručujúcu podobnosť s pôvodným obrázkom („s“ značí anglické slovo *similarity*). Stĺpec w_t určuje váhu funkcie na zvýšenie úspešnosti detekcie prechodov („t“ značí *transition*). Význam stĺpcov s úspešnosťou markerov tolerujúcich jednu, dve a tri chyby bude vysvetlený neskôr v tejto kapitole.

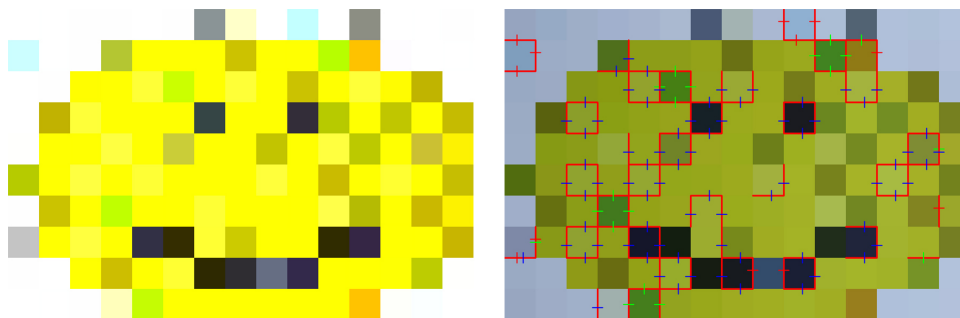


Obrázok 5.1: Obrázky z ktorých boli generované polia markerov: a) scenery, b) trenčín, c) vut, d) smile, e) stripes, f) colours

zdroj	farebný model	w_s	w_t	úspešnosť prechodov	úspešnosť markerov	úspešnosť markerov (1 chyba)	úspešnosť markerov (2 chyby)	úspešnosť markerov (3 chyby)
colours	YUV	0.5	0.5	83.88%	4.81%	10.58%	18.27%	24.04%
scenery	YUV	0.5	0.5	90.55%	7.69%	25%	42.31%	65.38%
trencin	YUV	0.5	0.5	95.15%	28.84%	50%	70.19%	80.77%
smile	YUV	0.6	0.4	87.27%	2.88%	5.77%	15.38%	29.81%
vut	RGB	0.7	0.3	90.91%	23.08%	38.46%	45.19%	64.42%
vut	YUV	0.8	0.2	96.12%	32.69%	55.77%	78.85%	90.38%
vut	YUV	0.5	0.5	96.24%	46.15%	65.36%	86.54%	95.19%

Tabuľka 5.1: Úspešnosti detekcie prechodov a markerov

Na obrázku 5.2 je vyobrazené vygenerované a rekonštruované pole markerov zo vstupného obrázku *smile*. Ide o jeden z najhorších výsledkov celého testovania. Počas procesu tlače a snímania došlo k výraznej zmene farieb, najmä k zníženiu vysokých intenzít a strate kontrastu. Snímanie v tomto konkrétnom prípade prebiehalo za denného svetla pomocou digitálnej zrkadlovky Canon EOS 1000D, ktorá prekvapujúco vykazovala horšie vyváženie bielej v porovnaní s menej kvalitnými fotoaparátmi v mobilných zariadeniach. Výsledkom je silný modrý nádych rekonštruovaného poľa markerov. Ďalším faktorom ktorý negatívne ovplyvnil výslednú úspešnosť bola samotná tlač. Vo vytlačenom poli markerov sa niektoré farby subjektívne odlišujú od svojej predlohy viac ako iné. Úspešnosť detekcie prechodov je v tomto prípade 87%, výsledná úspešnosť detekcie markerov iba 2.88%. Tak výrazný rozdiel medzi týmito dvomi číslami je spôsobený tým, že jeden chybné detekovaný prechod môže zapríčiniť v závislosti na svojej polohe až šesť nesprávne detekovaných markerov. Pritom prechodov je v tomto konkrétnom prípade 825, zatiaľ čo markerov iba 104. Faktorom ovplyvňujúcim percento úspešne detekovaných markerov nie je iba počet správnych prechodov, ale aj ich rozmiestnenie. V tabuľke 5.1 si je možné všimnúť, že aj napriek vyššej úspešnosti prechodov u poľa markerov *colours* oproti *smile* je percento správne detekovaných markerov nižšie.

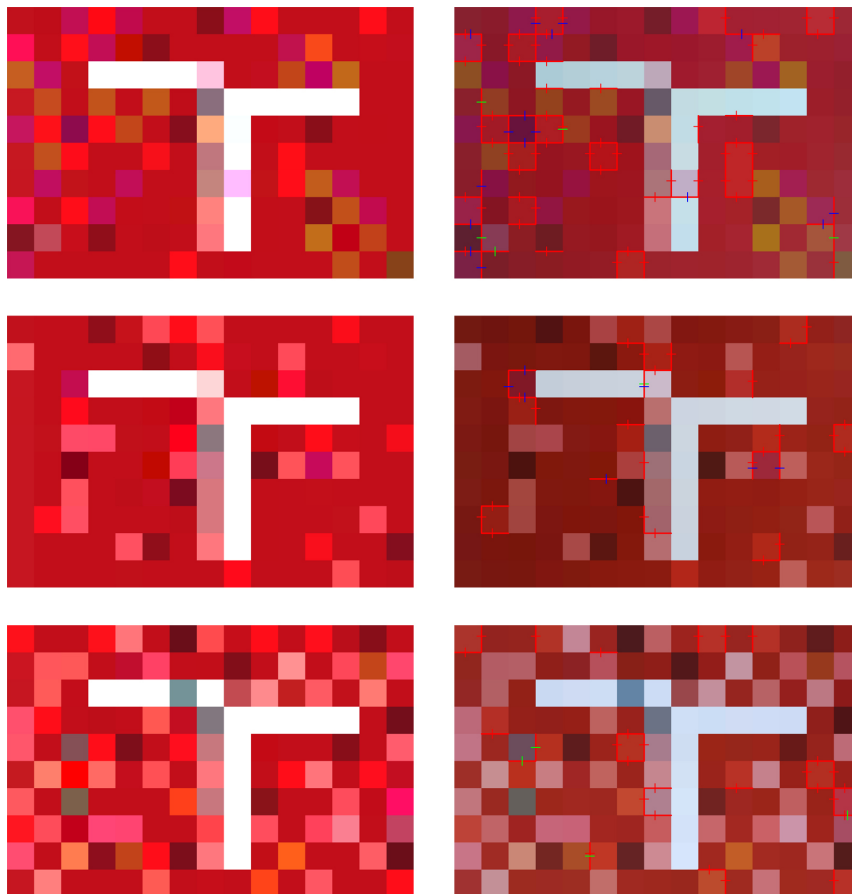


Obrázok 5.2: Vygenerované a rekonštruované pole markerov zo vstupného obrázku *smile*, $w_s = 0.6$, $w_t = 0.4$

O niečo lepšie výsledky dosiahli polia markerov vygenerované zo snímky *hut*. Dvojice vygenerovaných a rekonštruovaných polí markerov sú zobrazené na obrázku 5.3 po rade s takýmto nastavením parametrov hodnotiacej funkcie:

- hodnotenie podobnosti pomocou RGB modelu, $w_s = 0.7$, $w_t = 0.3$
- hodnotenie podobnosti pomocou YUV modelu s favorizovaním zmien v kanáli Y, $w_s = 0.8$, $w_t = 0.2$
- hodnotenie podobnosti pomocou YUV modelu s favorizovaním zmien v kanáli Y, $w_s = 0.5$, $w_t = 0.5$

Najvyššia úspešnosť bola dosiahnutá v poslednom prípade, až 96.24% správne detekovaných prechodov, veľmi podobný výsledok bol dosiahnutý aj v druhom prípade — 96.12%, pole markerov generované za použitia RGB farebného modelu dosiahlo 90.91% úspešnosť. Na obrázku 5.3 si je možné všimnúť dopad použitia RGB a YUV modelu na generované farby. Zatiaľ čo pri hodnotení podobnosti pomocou RGB dochádzalo k zmenám najmä v chrominancii farieb, za použitia modelu YUV sa menila najmä ich intenzita. Subjektívne sa môže javiť druhý prípad ako menej rušivý, navyše generovanie za použitia YUV spravidla



Obrázok 5.3: Dvojice vygenerovaných a rekonštruovaných polí markerov: *vt_RGB-0.7-0.3*, *vt_YUV-0.8-0.2*, *vt_YUV-0.5-0.5*

dosahovalo vyšších úspešností detekcie. Obrázok ďalej demonštruje vplyv nastavenia váh w_s a w_t na výsledné pole markerov. Podľa očakávania mali vyššie hodnoty váhy w_s pozitívny vplyv na podobnosť výsledného poľa markerov so zdrojovým obrázkom. Prekvapujúco však nie vždy dosahovali nižších úspešností v porovnaní s vysokými hodnotami váhy w_t . Toto bolo evidentné pokiaľ sa v zdrojovom obrázku nachádzali plochy konštantnej farby. V prípade vysokej váhy w_s sa na týchto plochách často vygenerovali dlaždice s rovnakou farbou, čo malo za následok ich bezproblémové detekovanie po vytlačení a nasnímaní. Ako bolo spomenuté v predchádzajúcom texte, pri detailnom snímaní malej časti poľa markerov by to mohlo mať negatívny vplyv na presnosť nájdania mriežky. Na druhú stranu, vysoké nastavenie váhy w_t preferuje výrazné hrany, čo sa prejavovalo zmenou dlaždíc na plochách konštantnej farby. A to v konečnom dôsledku mierne znížilo úspešnosť detekcie prechodov oproti dlaždiciam s rovnakou farbou.

Problém správnej detekcie markerov, ktoré prešli procesom tlače a snímania, sa dá chápať ako problém správneho prenosu kódového slova cez kanál vystavený pôsobeniu šumu. Jednou z možností zlepšenia úspešnosti detekcie markerov je zvýšenie Hammingovej vzdialenosti medzi nimi. Ako bolo spomenuté v predchádzajúcom texte, marker je z implementačného pohľadu definovaný ako vektor 36 znakov. Každé konkrétne pole markerov sa dá chápať ako kód, ktorý je určený súborom kódových slov — markerov. V pôvodnej implemen-

tácii sú generované kódy s Hammingovou vzdialenosťou rovnou 1, teda kódové slová dvoch rozdielnych markerov sa líšia minimálne v jednom znaku. V tom prípade chyba v jednom znaku (teda jeden nesprávne detekovaný prechod) môže mať za následok zámenu jedného kódového slova za druhé (teda jedného markeru za druhý). V prípade zvýšenia Hammingovej vzdialenosti kódu je možné detekovať a opravovať takéto chyby. Na detekciu k chýb je potrebná vzdialenosť $k + 1$, na opravu $2k + 1$.

Oproti klasickému kódovaniu má tento prístup jednu odlišnosť. Určitá Hammingova vzdialenosť sa nedosahuje predĺžením kódovaných slov o kontrolné znaky, ale vylúčením niektorých slov z kódu tak, aby bola dodržaná želaná vzdialenosť. Z implementačného pohľadu by sa Hammingova vzdialenosť d kódu zaručovala prehlásením dvoch markerov za konfliktné v prípade, že by sa líšili v menej ako d prechodoch intenzít.

Program *transitionDetector* bol upravený tak, aby každému markeru priradil počet chybných detekovaných prechodov a bol tak schopný vypočítať teoretickú úspešnosť detekcie markerov pre rôzne Hammingové vzdialenosti. Výsledky sú zobrazené v tabuľke 5.1 v posledných troch stĺpcoch. Nárast úspešnosti je značný, pre Hammingovu vzdialenosť dostatočnú na opravu troch chýb vystúpila v niektorých prípadoch nad 90%. Zaujímavým spôsobom sa vyvíja úspešnosť u vyššie spomínanej dvojice *colours* a *smile*. Zatiaľ čo pre nula, jedna a dve chyby dosiahlo väčšiu úspešnosť prvé z menovaných polí markerov, v prípade troch chýb je situácia opačná. To je dôsledkom menej rovnomerného rozmiestnenia chybných prechodov u *colours*. Ich zhľukovanie spôsobilo vyššiu úspešnosť pri schopnosti opraviť nula alebo malé množstvo chýb. Zlepšenie úspešnosti však nie je so stúpajúcim množstvom opravitelných chýb také markantné.

5.2 Štúdia stálosti farieb v procese tlače a snímania

Keďže výsledky testovania neboli uspokojivé, bolo po dohode s vedúcim práce rozhodnuté zamerať sa na skúmanie príčin nízkej úspešnosti detekcie a na jej zvyšovanie. Táto podkapitola odhaľuje javy ovplyvňujúce stálosť farieb. Spolu s tou nasledujúcou predstavujú obecnú štúdiu možnosti použitia farieb na kódovanie informácie s odolnosťou voči rušivým vplyvom spôsobeným nedokonalosťou procesu tlače, snímania alebo rozličnými svetelnými podmienkami. Štúdia presahuje rámec zadania tejto diplomovej práce.

V predchádzajúcom texte boli spomenuté niektoré faktory komplikujúce úspešnú detekciu. Jedným z nich je zmena priemernej svetlosti a kontrastu. Tabuľka 5.2 porovnáva priemernú hodnotu intenzity \bar{x} a jej smerodajnú odchýlku σ pre vygenerované (stĺpec *originál*) a rekonštruované polia markerov. Snímanie prebiehalo pomocou dvoch zariadení, HTC One V a Canon EOS 1000D, za denného svetla (v tabuľke označené ako *deň*) a za svetla žiarovky (*noc*). Polia markerov boli vybrané tak, aby pokryli viaceré situácie, *colours* obsahuje veľké množstvo farieb od tmavých po svetlé, *trenčín* má vysokú priemernú intenzitu a *vut* naopak nízku.

zdroj	originál		HTC deň		HTC noc		Canon deň		Canon noc	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
clrs	136.71	78.72	106.91	65.59	109.34	60.15	105.16	58.70	87.59	53.97
tn	204.46	58.08	132.48	46.36	136.40	46.96	143.82	51.67	123.48	48.67
vut	91.14	63.68	85.55	54.76	88.37	48.22	83.26	53.52	72.28	47.78

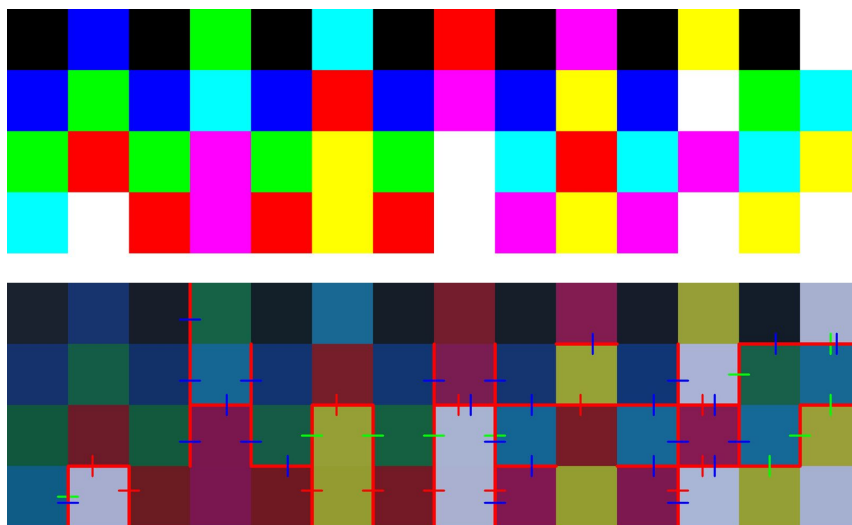
Tabuľka 5.2: Priemerná hodnota a smerodajná odchýlka intenzít u vygenerovaných a rekonštruovaných polí markerov, kde clrs je skratka *colours* a tn je skratka *trenčín*.

Pri všetkých meraniach došlo u rekonštruovaných polí markerov k zníženiu ako priemernej hodnoty, tak aj smerodajnej odchýlky. Najmarkantnejšie to bolo v prípade svetlého obrázku *trenčín*, u ktorého sa stredná hodnota znížila v priemere o 34%. Najmenší posun nastal u *vut*, necelých 10%.

Po vykonaní tohto experimentu boli „ručne“ zisťované zmeny u vybraných farieb. Výsledky viedli k hypotéze, ktorá tvrdí, že so zvyšujúcou sa intenzitou pôvodnej farby stúpa odchýlka, o ktorú bude táto farba tmavšia po jej vytlačení a nasnímaní. Tento trend je však iba približný, neplatí pre všetky testované prípady. Navyše, u veľmi tmavých farieb dochádza k miernemu zosvetleniu. Táto hypotéza bude podrobnejšie diskutovaná na inom mieste v tejto podkapitole.

Zníženie smerodajnej odchýlky u rekonštruovaných polí markerov oproti tým generovaným je pravdepodobne dôsledkom týchto vyšších zmien v intenzitách u svetlých a veľmi tmavých farieb. To spôsobí, že rozdiely medzi intenzitami nasnímaných farieb sú menšie, čiže aj výsledný kontrast rekonštruovaného obrázku je nižší.

Pre potreby tejto práce je však dôležitejšie zaoberať sa zmenami v jednotlivých farebných zložkách. Ako sa proces tlače a snímání podpíše na intenzitách v separátnych RGB kanáloch je vysvetlené pomocou obrázku 5.4. V jeho vrchnej polovici je pole farebných dlaždíc (nejedná sa o pole markerov) tak ako bolo vygenerované. Sú v ňom použité výlučne farby, ktoré majú hodnotu každého z troch RGB kanálov buď 0 alebo 255. Jedná sa teda o farby: čierna, biela, červená, zelená, modrá, cyan, magenta a žltá. Mohlo by sa javiť, že po vytlačení a nasnímaní obrázkov budú prechody medzi dlaždicami detekované bez chýb, keďže prechody v pôvodnom vygenerovanom obrázku mali buď nulovú hodnotu, alebo maximálnu, teda 255. Spodná časť obrázku dokazuje, že to tak nie je.



Obrázok 5.4: Vygenerované a rekonštruované pole farieb s intenzitami 0 alebo 255

Nasledujúci text vysvetľuje chybné detekovaný prechod v zelenom a modrom kanáli medzi bielou dlaždicou v pravom hornom rohu a dlaždicou farby cyan o riadok nižšie. Vo vygenerovanom obrázku sa v RGB modeli jedná o farby (255, 255, 255) a (0, 255, 255) so svetlosťou 255 a 179, v tomto poradí. Rekonštruované boli farby (167, 176, 207) a (23, 104, 147), s intenzitami 177 a 84. Tento konkrétny prípad je jednou z výnimiek vyššie uvedeného trendu, keďže zmena intenzity bielej (čiže svetlejšej) farby bola menšia ako zmena u farby

cyan. Zaujímavejšie však je, že zatiaľ čo u bielej farby došlo k poklesu intenzít v zelenom a modrom kanáli z 255 na 176 resp. 207, u farby cyan bol pokles vyšší, a to až na 104 resp. 147. A to práve malo za následok nesprávnu detekciu týchto dvoch prechodov. Vygenerované boli ako rovnaké ($255 \rightarrow 255$), rekonštruované boli ako klesajúce ($176 \rightarrow 104$ resp. $207 \rightarrow 147$). Táto situácia sa opakuje aj pri iných kombináciách farieb.

Experiment bol zopakovaný za použitia farieb zložených z menej okrajových intenzít, konkrétne 30 a 220 alebo 50 a 200. Výsledky boli podobné. Obecne sa dá povedať, že pokiaľ majú dve dlaždice vysokú intenzitu v jednom resp. dvoch rovnakých kanáloch a v tom jednom resp. dvoch zvyšných sa ich intenzity výrazne odlišujú, je veľmi pravdepodobné, že dôjde k zmene typu prechodu resp. prechodov.

Ďalším problémom pri snahe o stálosť farieb je fakt, že gamut tlačiarne nepokrýva celú množinu farieb definovateľnú pomocou RGB farebného modelu.

Spolu so spomenutými faktormi vstupuje do procesu rekonštrukcie farieb napr. vplyv rôzneho osvetlenia alebo správneho nastavenia snímacieho zariadenia. Analytický popis všetkých činiteľov ovplyvňujúcich stálosť farieb by bol veľmi komplexný a aj tak by sa pravdepodobne nevyhol istým zjednodušeniam. Preto bolo rozhodnuté pokračovať v štúdií praktickejšou cestou.

Ideou bolo vygenerovať, vytlačiť a nasnímať čo najväčší počet farieb a následne sledovať k akým zmenám u nich došlo. Farby boli generované ako štvorcové dlaždice v dvojrozmerných poliach, tak aby mohli byť s minimálnymi zmenami použité už implementované nástroje *MFGGenerator*, *gridDetector* a *transitionDetector*.

Do prvého spomenutého programu bola doimplementovaná trieda na generovanie polí dlaždíc s náhodnými farbami. Tie úmyselne neboli generované v nijakej postupnosti, ich náhodnosť bola dôležitá pre ďalšie experimenty, popísané v nasledujúcej podkapitole. Zvyšné dva programy boli použité bez zmien, *transitionDetector* nebol použitý na vyhodnocovanie úspešnosti ale výlučne na rekonštrukciu poľa dlaždíc.

Na samotné sledovanie zmien farieb bol vytvorený nový program, *colourTester*. Jeho vstupom je séria dvojíc obrázkov — vygenerovaných a rekonštruovaných polí dlaždíc. Program iteruje cez vstupné dvojice a pre každú dlaždicu vypočíta odchýlku jej rekonštruovanej farby oproti vygenerovanej a to v každom farebnom kanáli zvlášť. Táto informácia sa potom uloží do vnútorných dátových štruktúr programu, z ktorých je možné vygenerovať graf alebo exportovať textový záznam priemerných odchýlok.

Keďže pri použití 8 bitovej farebnej hĺbky na kanál je počet výsledných farieb väčší ako 16 miliónov, bolo by problematické počítať štatistiku zmien pre každú farbu zvlášť. Pri rozmeroch polí dlaždíc 15×10 by ich bolo nutné použiť viac ako 100 000. Navyše, na zvyšovanie vierohodnosti výsledkov je potreba získať viacero záznamov tej istej farby za rôznych podmienok v procese tlače a snímania.

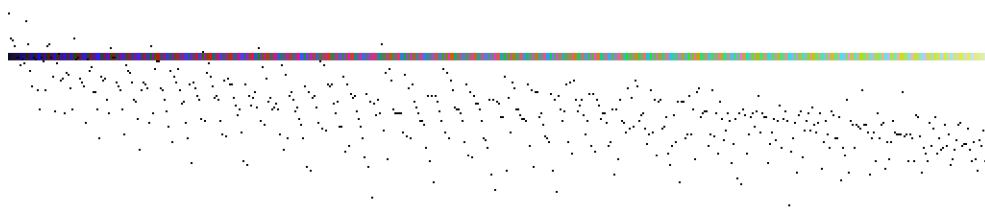
Z toho dôvodu nebola databáza vytváraná pre všetky farby, ale iba pre ich určité oblasti. Intenzity každej farebnej zložky boli rozdelené na n intervalov, oblasti farieb vznikli ich kombináciami. To malo za následok namapovanie celkového množstva 256^3 farieb do n^3 oblastí. Pri analýze zmien vygenerovanej farby bolo najskôr zistené do ktorej oblasti spadá, následne boli adekvátne upravené štatistiky danej oblasti.

Ako rozumným počtom intervalov sa ukázal $n = 8$, ktorý rezultuje v $8^3 = 512$ farebných oblastí. Napríklad, farba (50, 238, 154) bude namapovaná do ($\langle 32, 63 \rangle$, $\langle 224, 255 \rangle$, $\langle 128, 159 \rangle$).

Databáza bola vytvorená za použitia 83 dvojíc vstupných obrázkov, ktoré spolu obsahovali 24659 dlaždíc. V prípade použitia $n = 8$ intervalov pripadá na jednu oblasť farieb v priemere 48 záznamov. Snímanie prebiehalo v rôznych svetelných podmienkach a za použitia všetkých zariadení spomenutých na začiatku tejto kapitoly. Na tlač boli použité dve

rozdielne tlačiarne.

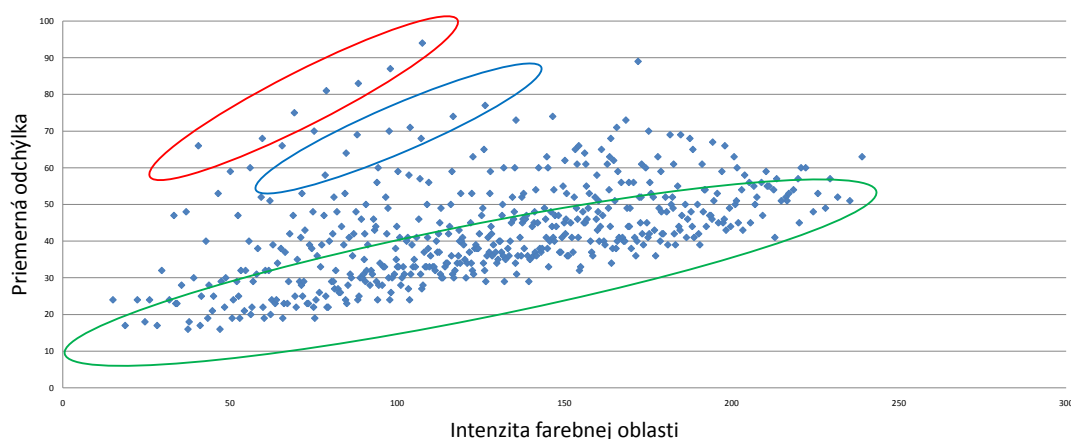
Na obrázku 5.5 je graf vygenerovaný priamo programom *colourTester*. Databáza bola vytvorená pre $n = 8$ intervalov, každá z rezultujúcich 512 oblastí je reprezentovaná na vodorovnej osi farbou, ktorá zodpovedá jej stredu. Zoradené sú vzostupne podľa svetlosti. Vzdialenosť funkčných hodnôt od vodorovnej osi určuje priemerný rozdiel rekonštruovanej farby od vygenerovanej. Hodnoty nad osou značia, že došlo k zosvetleniu farby, body pod ňou značia stmavenie. Úlohou grafu je iba vystihnúť trend odchýlok, nie ich presne definovať. Z toho dôvodu v ňom nie sú uvedené presné hodnoty, ani nie je naznačená zvislá os. Navyše, funkčné hodnoty sa počítali ako aritmetický priemer zmien vo všetkých troch RGB kanáloch. Tieto zmeny boli ukladané so znamienkom, preto mohlo v niektorých prípadoch výpočtu priemeru dôjsť k ich čiastočnému vyrušeniu.



Obrázok 5.5: Priemerná zmena farby v procese tlače a snímania

Graf potvrdzuje vyššie uvedenú hypotézu, ktorá tvrdí, že u veľmi tmavých farieb dochádza k zosvetleniu, u ostatných k stmaveniu, ktoré má so stúpajúcou intenzitou zvyšujúci sa trend.

Graf na obrázku 5.6 vznikol z rovnakých dát ako ten predchádzajúci, akurát odchýlky v jednotlivých kanáloch neboli počítané so znamienkom, ale ako absolútne hodnoty. To má za následok, že všetky priemerné odchýlky v grafe sú kladné. Vynesené sú na zvislú os, tá vodorovná popisuje intenzitu stredu farebnej oblasti. Zaujímavé časti grafu sú označené farebnými elipsami.



Obrázok 5.6: Priemerná absolútna zmena farby v procese tlače a snímania

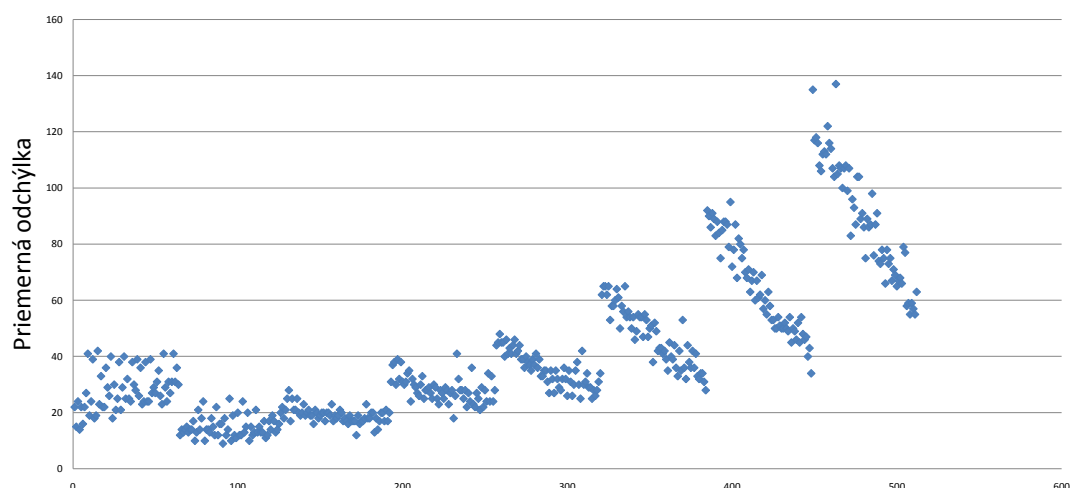
Tá červená popisuje zhluk hodnôt s veľmi vysokou priemernou odchýlkou medzi vyge-

nerovanou a rekonštruovanou farbou. Všetky body v tomto zhluku spája jedna vlastnosť. Reprezentujú farebné oblasti, ktorých modrý kanál patrí do intervalu $\langle 224, 255 \rangle$, zelený do $\langle 0, 31 \rangle$ a červený sa zľava doprava mení od najnižších hodnôt po tie najvyššie. Príčina tohto javu bola odhalená pomocou programu *Adobe Photoshop* a jeho funkcie s názvom *Gamut Warning*. Tá na základe nastaveného ICC profilu pre CMYK (v tomto prípade „U.S. Web Coated (SWOP v2)“) dokáže určiť, či je daná farba pokrytá gamutom tlačiarne, ktorá tento profil používa. Všetky farebné oblasti prislúchajúce bodom v červenej elipse boli označené za netlačiteľné. Po použití funkcie na korekciu do gamutu tlačiarne bola napríklad farba (239, 15, 239), ktorá je približne stredom oblasti ($\langle 224, 255 \rangle$, $\langle 0, 31 \rangle$, $\langle 224, 255 \rangle$), zmenená na (181, 79, 157). To znamená priemernú zmenu na úrovni 68 a to ide iba o skreslenie spôsobené nedokonalosťou procesu tlače. V grafe prislúcha tejto farebnej oblasti odchýlka 94.

Pre body v modrej elipse tiež platí, že ležia mimo gamut tlačiarne. V porovnaní s predchádzajúcim zhlukom sa však intenzita v zelenom kanáli posunula do intervalu $\langle 32, 63 \rangle$, pre ostatné kanále zostáva situácia rovnaká. Pre body nepatriace ani do jednej elipsy platí, že väčšina z nich je mimo gamut.

Opačný prípad nastal pre body v zelenej elipse. Väčšina z nich prislúcha oblastiam, ktoré sú tlačiteľné. Na týchto bodoch je dobre vidieť trend stúpajúcej odchýlky pri stúpajúcich hodnotách intenzity farieb.

Pre potreby tejto práce je však dôležitejšie zistiť, ako sa počas procesu tlače a snímania mení intenzita v jednotlivých kanáloch. Graf na obrázku 5.7 zobrazuje priemerné odchýlky pre zelený kanál. Tak ako v predchádzajúcom prípade sú použité absolútne hodnoty zmien, preto sú všetky hodnoty kladné. Na vodorovnú os sú vynesené merania zodpovedajúce všetkým farebným oblastiam, je ich teda 512. Merania sú primárne zoradené podľa vzostupnej intenzity v zelenom kanáli, sekundárne podľa vzostupnej svetlosti celej farebnej oblasti ktorú reprezentujú. Na obrázku sa to prejavilo vytvorením ôsmich zhlukov, v každom z nich sú oblasti s konštantnou intenzitou v zelenom kanáli, $\langle 0, 32 \rangle$ pre najľavejší, $\langle 224, 255 \rangle$ pre najpravejší. Keďže je každý zhluk usporiadaný podľa stúpajúcej svetlosti, najľavejší bod zhluku reprezentuje najtmavšiu farebnú oblasť pre danú hodnotu zeleného kanála, najpravejší bod predstavuje najsvetlejšiu.



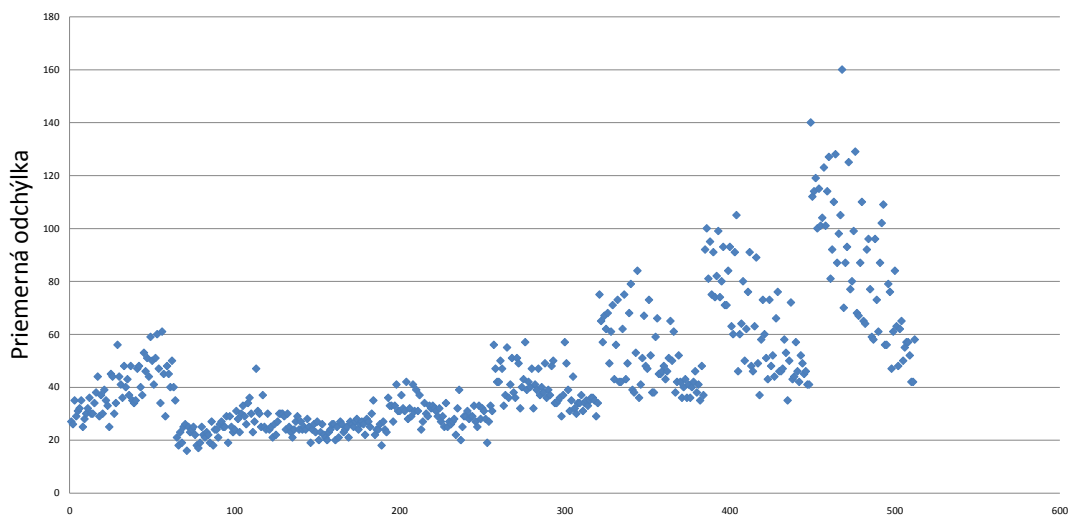
Obrázok 5.7: Priemerná odchýlka v zelenom kanáli

Z grafu sa dá vyčítať viacero faktov. Najnižšie odchýlky boli dosiahnuté u druhého a tretieho zhluku, teda pre intenzity zeleného kanála v intervaloch $\langle 32, 63 \rangle$ a $\langle 64, 95 \rangle$. V rámci týchto zhlukov bol zaznamenaný aj najnižší rozptyl odchýlok. Ten sa zvyšoval so zvyšujúcou sa intenzitou zelenej farby. U posledných troch zhlukov je viditeľné diagonálne rozmiestnenie jednotlivých odchýlok.

Nasledujúci text popisuje posledný zhluk. Väčšina farebných oblastí je mimo tlačiteľný gamut. Pre oblasti reprezentované najľavejšími bodmi obecnne platí, že majú vysokú intenzitu v zelenom kanáli a nízke intenzity vo zvyšných dvoch. Táto situácia má za následok vysokú odchýlku u zelenej farebnej zložky, zatiaľ čo odchýlky pre červenú a modrú sú pomerne nízke. Napr. pre oblasť $(\langle 32, 63 \rangle, \langle 224, 255 \rangle, \langle 32, 63 \rangle)$ boli namerané odchýlky 20, 106, 48 postupne pre jednotlivé kanále, čo rezultuje v priemernú odchýlku na hodnote 58. Naopak, najpravejšie body predstavujú oblasti s vysokými intenzitami vo všetkých troch farbách. To má za následok zníženie odchýlky v zelenom kanáli, napr. pre oblasť $(\langle 160, 191 \rangle, \langle 224, 255 \rangle, \langle 224, 255 \rangle)$ boli vypočítané odchýlky 52, 66, 52 v tomto poradí. Oproti predchádzajúcemu prípadu sa zmena v zelenom kanáli prudko znížila, v ostatných dvoch došlo k jej zvýšeniu. Priemerná chyba cez všetky tri kanále však zostala približne rovnaká: 57. Ďalšie merania potvrdili trend, že so stúpajúcou intenzitou červenej a modrej farby sa odchýlka distribuuje rovnomernejšie do jednotlivých kanálov.

Dve najvyššie hodnoty v tomto zhluku patria oblastiam $(\langle 0, 31 \rangle, \langle 224, 255 \rangle, \langle 224, 255 \rangle)$ a $(\langle 0, 31 \rangle, \langle 224, 255 \rangle, \langle 0, 31 \rangle)$. Ide o hodnoty silne nad rámec gamutu tlačiarne, prvá z nich patrila v grafe 5.6 do červenej elipsy.

Analogický trend bol zaznamenaný aj pre modrý kanál, jeho graf sa nachádza na obrázku 5.8. Rozdielom sú o niečo „širšie“ zhluky hodnôt a vyššia priemerná aj maximálna odchýlka.

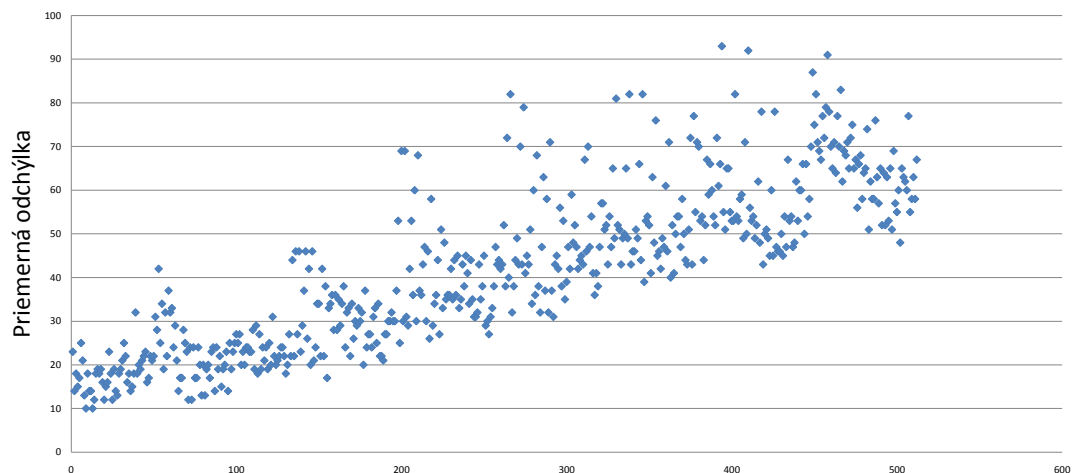


Obrázok 5.8: Priemerná odchýlka v modrom kanáli

Mierne odlišná situácia nastáva pre červený kanál, popísaný grafom na obrázku 5.9. Zhluky meraní už nie sú tak dobre odlišiteľné, najnižšie zmeny intenzít červenej farby boli dosiahnuté u prvého z nich.

Tabuľka 5.3 zobrazuje štatistiky odchýlok pre posledné tri grafy. Najmenej stály sa ukázal byť modrý kanál a to vo všetkých ukazovateľoch. Najnižšia priemerná odchýlka a medián boli namerané u zelenej farby, červená dosiahla najnižšiu maximálnu hodnotu zmeny intenzity.

Textový súbor s výsledkami štúdie sa nachádza v adresári `\study\singleColor` na pri-



Obrázok 5.9: Priemerná odchýlka v červenom kanáli

kanál	min	max	priemer	medián
červený	10	93	41.84	42
zelený	9	137	40.99	32.5
modrý	16	160	45.05	37

Tabuľka 5.3: Štatistika odchýlok pre jednotlivé farebné kanále

loženom disku DVD. Súbor je formátovaný tak, aby ho bolo možné jednoducho importovať do iných programov.

5.3 Štúdia úspešnosti detekcie prechodov u dvojíc farieb

Poznatky z predchádzajúcej kapitoly mohli byť zapracované do algoritmu na generovanie polí markerov za účelom zvýšenia úspešnosti detekcie. Reprezentácia markerov pomocou intenzitných prechodov však zapríčiňuje, že úspešnosť sa de facto neodvíja od zmien farieb samotných, ale od toho ako sa menia farby susedných dlaždíc jedna voči druhej. Z teoretického pohľadu by na úspešnosť nemalo vplyv, ak by počas procesu tlače a snímání došlo k zmene farieb o nejakú hodnotu, pokiaľ by táto hodnota bola pre všetky dlaždice rovnaká. Prechody intenzít by sa v tomto prípade nezmenili.

Z tohto dôvodu bolo rozhodnuté nadviazať na predchádzajúcu štúdiu ďalšou, ktorá si kladie za cieľ určiť pravdepodobnosť správnej detekcie prechodov pre rôzne dvojice farieb. Tento proces prebieha automatizovane. Vstupom do programu sú opäť dvojice vygenerovaných a rekonštruovaných polí farieb prípadne polí markerov. Program postupuje tak, že prechádza každú dvojicu dlaždíc (tie nemusia byť iba susedné) vo vygenerovanom poli a vypočíta, aké RGB prechody táto dvojica reprezentuje. Následne spraví to isté pre odpovedajúce farby rekonštruovaného poľa a porovná, či sú prechody rovnakého typu. Posledným krokom je upravenie štatistík úspešnosti detekcie pre dvojicu farieb z vygenerovaného poľa. Tak ako v predchádzajúcej štúdii sa kvôli veľkému počtu farieb rozdelí každý farebný kanál do k intervalov, z čoho rezultuje k^3 oblastí. Počet jedinečných dvojíc týchto oblastí

vyjadruje vzorec 5.1.

$$n = \frac{k^3 \times k^3}{2} + \frac{k^3}{2} \quad (5.1)$$

Pre $k = 8$, čo sa javí pri objeme testovacích dát ako optimálne číslo, je počet farebných oblastí 512, z čoho rezultuje 131328 kombinácií. Oproti predchádzajúcej štúdii je toto číslo podstatne väčšie, avšak tým, že sa v rámci jedného poľa farieb porovnáva každá dlaždica s každou, sa počet meraní zvyšuje. Napríklad, pre pole o veľkosti 15×10 sa vykoná viac ako 11 tisíc porovnaní. Nie je možné porovnávať farby z rôznych polí navzájom, keďže tie mohli byť vytlačené alebo nasnímané za odlišných podmienok. Za použitia rovnakej testovacej sady ako v predchádzajúcej štúdii sa podarilo získať viac ako 4 milióny meraní, čiže na jednu dvojicu oblastí farieb pripadá v priemere 31 záznamov.

Pre tieto účely bola implementácia programu *colourTester* rozšírená o tri nové triedy. Tie umožňujú vytvorenie databáze úspešností dvojíc oblastí farieb z testovacích dát ako aj jej exportovanie do textového súboru. Aby bolo možné získané údaje použiť pri generovaní polí markerov a tým pádom zvýšiť úspešnosť detekcie, umožňujú tieto triedy načítať databázu z exportovaného súboru a vrátiť pravdepodobnosť správnej detekcie prechodov pre danú dvojicu farieb. Tejto problematike sa bližšie venuje nasledujúca podkapitola.

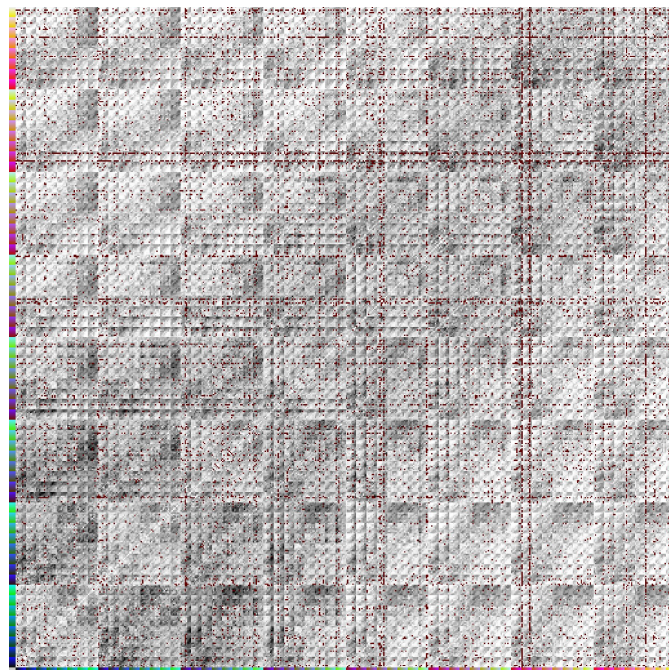
Veľkou výhodou tohto riešenia je, že je založené na praktickom pozorovaní skutočnosti. Pôvodný plán bol implementovať hodnotiacu funkciu genetického algoritmu tak, aby defavorizovala farby, ktoré sú mimo gamut tlačiarne. To je však iba jeden z faktorov ovplyvňujúcich výslednú úspešnosť. Vyššie spomenutý prístup už zo svojej podstaty zahŕňa všetky činitele vstupujúce do procesu tlače a snímania.

Výsledky štúdie demonštruje obrázok 5.10. Na horizontálnu aj vertikálnu os sú vynesené stredy všetkých 512 farebných oblastí. Zoradené sú zľava doprava, resp. zdola nahor tak, ako by ich vygenerovali tri zanorené cykly, pričom najvrchnejší by iteroval cez oblasti červenej farby, stredný cez zelené a najviac zanorený cez modré oblasti. Prvá osmina oboch osí teda zodpovedá oblastiam s červeným kanálom v intervale $\langle 0, 31 \rangle$, v druhej osmine je to $\langle 32, 63 \rangle$, atď. Analogicky je každá osmina rozdelená na osem úsekov s konštantou zelenou zložkou. Na najnižšej úrovni sa iteruje cez oblasti modrej farby. Graf je funkciou dvoch premenných, pričom oborom hodnôt sú percentuálne úspešnosti správnej detekcie prechodov vyjadrené odtieňom šedej. Svetlá farba značí vysokú úspešnosť, tmavá nízku. Červená farba v grafe vyjadruje, že pre danú kombináciu farebných oblastí neboli odmerané žiadne hodnoty. Graf je súmerný podľa pomyslenej priamky $x = y$.

Z grafu je možné vyčítať niekoľko skutočností. Vysoká úspešnosť bola dosiahnutá v prípade, že sa porovnávali farby z rovnakých oblastí. To sa v grafe prejavilo svetlou oblasťou na diagonále $x = y$. Tento jav sa dá interpretovať tak, že veľmi podobné farby sa zmenia v procese tlače a snímania podobným spôsobom.

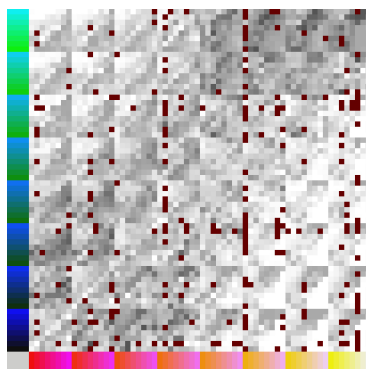
Ďalšou charakteristickou črtou je, že širšie okolie diagonály je v priemere tmavšie ako ľavý hodný a pravý dolný roh. K tomuto lokálnemu stmaveniu, a teda zníženiu úspešností, došlo v dôsledku malej vzdialenosti v červenom kanáli medzi príslušnými farebnými oblasťami. Kvôli tomu už menšie rušivé vplyvy v procese tlače a snímania môžu spôsobiť detekovanie rozdielneho typu prechodu. To, že tento jav nie je v grafe pre ostatné dva kanále viditeľný v takomto rozsahu, je spôsobené práve usporiadaním farebných oblastí na osiach. Tým, že červená farba sa mení s najmenšou frekvenciou sú v obraze veľké plochy s rovnakou intenzitou v tomto kanáli.

Zelená a modrá farebná zložka ovplyvňujú graf v menšom meradle. Na obrázku 5.11 je zobrazený výrez z pravého dolného rohu pôvodného grafu, zvislá os bola dokreslená



Obrázok 5.10: Graf úspešností dvojíc farebných oblastí

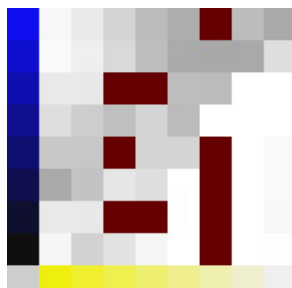
dodatočne. Na každú os je vynesných 64 farebných oblastí, intenzity v červenom kanáli sú konštantné, na vodorovnej osi v intervale $\langle 224, 255 \rangle$, na zvislej $\langle 0, 31 \rangle$. V obrázku sa dajú rozoznať štvorcové oblasti so stranou dĺžky 8. Pre ne je charakteristická konštantná intenzita v zelenom kanáli. Opäť je možné pozorovať určité stmavenie v oblasti okolo pomyselnéj priamky $x = y$, tento raz je to však spôsobené malou vzdialenosťou v zelenom kanáli. Situácia je analogická k tej, ktorá sa týkala červenej farebnej zložky a celého grafu.



Obrázok 5.11: Výrez z grafu úspešností dvojíc farieb zobrazujúci oblasť konštantnej intenzity v zelenom kanáli

Na obrázku 5.12 je ešte podrobnejší pohľad na graf, ide o výrez pravej dolnej časti z obrázku 5.11. V grafe sa menia iba intenzity v modrej farebnej zložke, červená a zelená spadajú do intervalu $\langle 224, 255 \rangle$ na vodorovnej ose, resp. $\langle 0, 31 \rangle$ na zvislej ose. Vysoká úspešnosť v pravej dolnej časti je spôsobená veľkými rozdielmi medzi farebnými oblasťami.

Na jednej strane je maximálna intenzita v červenom a zelenom kanáli a vysoká intenzita v modrom, na druhej strane sú minimálne intenzity v prvých dvoch spomínaných kanáloch a nízka v treťom. Podobná situácia je aj v ľavom hornom rohu. V predchádzajúcej podkapitole bolo spomínané, že vysoká pravdepodobnosť nesprávnej detekcie prechodov nastáva aj v takom prípade, keď majú všetky tri farebné zložky jednej farby vysoké intenzity, zatiaľ čo u druhej farby je vysoká intenzita iba v jednom alebo dvoch kanáloch. Takýto prípad nastáva v pravom hornom rohu obrázka. Porovnáva sa farebná oblasť s vysokými intenzitami vo všetkých kanáloch s oblasťou, ktorá má silne zastúpenú iba modrú zložku. Následkom je, že intenzita v modrom kanáli sa u druhej farebnej oblasti zníži podstatne viac ako u prvej a to spôsobí zmenu typu prechodu.



Obrázok 5.12: Výrez z grafu úspešností dvojíc farieb zobrazujúci oblasť konštantnej intenzity v modrom kanáli

Graf v plnom rozlíšení spolu s textovým výpisom úspešností pre jednotlivé dvojice oblastí sa nachádza v adresári `\study\colorPairs` na priloženom disku DVD.

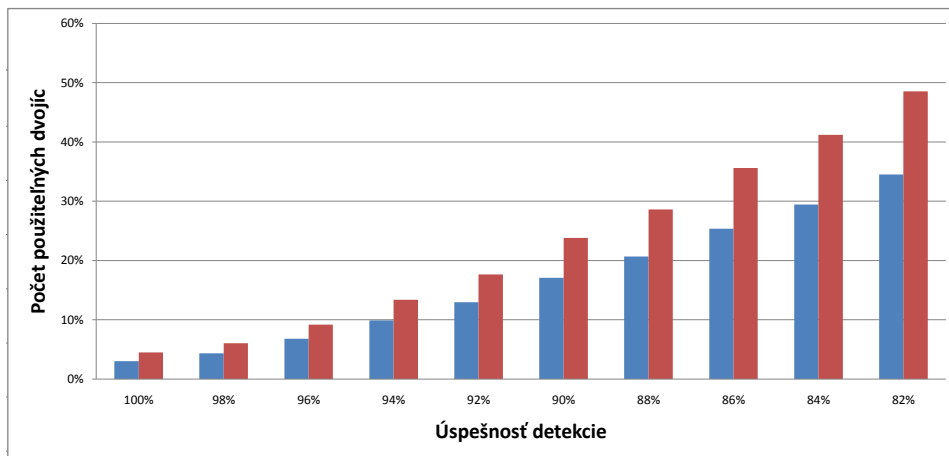
Tabuľka 5.4 zobrazuje počet párov farieb, ktoré dosiahli určitú úspešnosť detekcie prechodov. Napríklad, pri želanej 90% úspešnosti by bolo možné použiť iba približne každú piatu dvojicu. Situáciu ďalej zhoršuje fakt, že v poli markerov je každá dlaždica (s výnimkou tých okrajových) súčasťou štyroch párov, takže generovanie farebných dvojíc nie je nezávislé od okolia.

úspešnosť detekcie	počet použiteľných dvojíc	percentuálny počet použiteľných dvojíc
100%	3968	3.02%
98%	5718	4.35%
96%	8922	6.79%
94%	9371	9.92%
92%	17011	12.95%
90%	22447	17.09%
88%	27157	20.68%
86%	33330	25.38%
84%	38671	29.45%
82%	35356	34.54%

Tabuľka 5.4: Počet použiteľných dvojíc farebných oblastí pre rozdielne úspešnosti detekcie

Počet použiteľných farebných dvojíc sa dá zvýšiť uplatnením nasledujúcej myšlienky. Ak sa u nejakej dvojice farieb vo veľkej väčšine prípadov zmenil typ prechodu v nejakom kanáli na jeden iný konkrétny typ, je možné túto zmenu vo fáze generovania predpovedať. Dvojicu farieb je možné použiť v poli markerov s tým, že program rozpoznávajúci markery nebude očakávať vygenerovaný typ prechodu, ale ten predpovedaný.

Táto myšlienka bola zapracovaná do programu *colourTester*, výsledky sú zobrazené v grafe 5.13. Na vodorovnej osi je želaná úspešnosť detekcie, na zvislej je percentuálne vyjadrenie množstva použiteľných dvojíc farieb. Modré stĺpce predstavujú počet takýchto dvojíc pred aplikáciou vyššie spomenutej myšlienky, červené stĺpce po nej. Nárast počtu použiteľných párov sa pohybuje medzi 30% a 40%.



Obrázok 5.13: Graf použiteľných dvojíc farebných oblastí pre rôzne úspešnosti. Modré stĺpce predstavujú počet dvojíc pred aplikovaním myšlienky o predpovedi typu prechodov, červené stĺpce po ňom.

5.4 Využitie získaných poznatkov na zvýšenie úspešnosti detekcie

Poznatky o úspešnosti detekcie pre rôzne dvojice farieb boli zapracované do algoritmu na generovanie polí markerov. Ako bolo spomenuté v predošlom texte, trieda na vytváranie štatistík je schopná exportovať úspešnosti do textového súboru a tiež ich z neho načítať. Vďaka tomu mohla byť bez zmien použitá v programe *MFGenerator*.

V prípade, že bude pomocou parametra príkazového riadku špecifikovaný súbor obsahujúci štatistiky dvojíc farebných oblastí, bude na tieto informácie prihliadané počas generovania polí markerov. Implementované je to ďalšou čiastkovou funkciou hodnotiacej funkcie genetického algoritmu. Skóre každej dlaždice sa vypočíta ako priemerná hodnota úspešnosti detekcie medzi ňou a každou z jej susedných dlaždíc. Výsledok je následne vážený a spolu s výstupmi ostatných čiastkových funkcií tvorí hodnotenie danej dlaždice.

Porovnanie úspešnosti detekcie za použitia databázy dvojíc farebných oblastí a bez nej je uvedené v tabuľkách 5.5, 5.6 a 5.7. Testy boli vykonané za použitia rovnakých zariadení a v rovnakých svetelných podmienkach ako v predchádzajúcom prípade, popísanom tabuľkou 5.2. Dvojice polí markerov, za použitia databázy a bez nej, boli generované s takými parametrami, aby sa subjektívne rovnako podobali na predlohu. Skratka *db* uvedená

za názvom poľa markerov značí, že bola použitá databáza, v prípade *country db ** boli váhy jednotlivých čiastkových hodnotiacich funkcií genetického algoritmu nastavené tak, aby sa silne preferovala vysoká úspešnosť detekcie.

pole markerov	HTC noc	HTC deň	Canon noc	Canon deň	priemer
colours db	91.76%	93.58%	91.75%	92.48%	92.39%
colours	83.39%	83.87%	79.88%	85.09%	83.06%
country db	90.67%	91.64%	91.39%	92.85%	91.64%
country db *	95.03%	96.24%	94.9%	96.6%	95.69%
country	88.36%	90.55%	87.76%	89.21%	88.97%
trenčín db	96.12%	98.06%	95.88%	96.48%	96.64%
trenčín	89.21%	95.15%	90.9%	91.63%	91.72%
vut db	98.42%	98.9%	98.79%	99.27%	98.85%
vut	94.64%	96.12%	93.27%	95.55%	94.9%

Tabuľka 5.5: Úspešnosť detekcie prechodov

Tabuľky ukazujú, že prístup využívajúci databázu úspešností dosiahol lepšie výsledky vo všetkých prípadoch. Zatiaľ čo u detekcie prechodov sa zlepšenie nemusí javiť ako veľmi výrazné, rozdiely v úspešnostiach markerov sú značne rozdielne. Je to spôsobené už spomínaným faktom, že jeden nesprávne detekovaný prechod môže zapríčiniť chybu až v šiestich markeroch. Preto aj malé zvýšenie úspešnosti detekcie prechodov má za následok prudké zlepšenie výsledkov pre markery.

pole markerov	HTC noc	HTC deň	Canon noc	Canon deň	priemer
colours db	6.73%	8.65%	2.88%	5.77%	6.01%
colours	0.96%	4.81%	0.96%	0.96%	1.92%
country db	4.8%	6.73%	0.96%	3.85%	4.09%
country db *	25.96%	27.88%	18.27%	22.12%	23.56%
country	4.81%	7.69%	3.85%	7.69%	6.01%
trenčín db	42.31%	50.96%	25%	32.69%	37.74%
trenčín	0.96%	28.85%	1.92%	7.69%	9.86%
vut db	82.69%	80.77%	82.69%	85.58%	82.93%
vut	23.08%	32.69%	17.31%	24.88%	24.49%

Tabuľka 5.6: Úspešnosť detekcie markerov pri oprave 0 chýb

Aj za použitia databáze dvojíc farieb sa výsledky detekcie markerov môžu zdať neuspokojivé. Tie však nemusia odpovedať úspešnosti lokalizácie kamery v reálnej aplikácii využívajúcej rozšírenú realitu. Tá sa môže v závislosti na implementácii algoritmu, ktorý určuje, aká časť poľa markerov je snímaná kamerou, zvyšovať. Úspešnosť detekcie markerov by jej odpovedala v najhoršom možnom prípade, a to vtedy, keď je kamerou snímaný jediný marker.

V nasledujúcom texte je použitý pojem *transformovaný marker*. Myslí sa ním taký marker, ktorý sa v dôsledku chybných prechodov intenzit zmenil na iný, avšak existujúci marker. Jeho detekcia teda môže viesť k nesprávne určeniu pozície snímacieho zariadenia. V situácii, kedy je v zábere kamery väčšie množstvo markerov, pričom časť z nich je rozpoznaná správne, niektoré nie sú rozpoznané vôbec a zvyšné sú transformované, by mohol algoritmus postupovať nasledovne. Jeho cieľom by bolo rozlíšiť transformované markery od tých správ-

nych. Riešením by mohol byť odhad pozície kamery pre každý marker a následné vylúčenie tých, pre ktoré bola odhadnutá pozícia výrazne odlišná od tej najčastejšie sa vyskytujúcej. Je málo pravdepodobné, že by bola podobná pozícia kamery vypočítaná pre také množstvo transformovaných markerov, aby bola vyhlásená za modus. Z tohto dôvodu by tento prístup mohol v praxi fungovať.

Ďalšou možnosťou je sledovanie pohybu kamery a odhad jej trajektórie pre budúce snímky.

Z tabuľky 5.7 vyplýva, že úspešnosť detekcie markerov pri schopnosti opraviť štyri chyby v prechodoch medzi dlaždicami sa zvýšila. Za použitia databázy bola priemerná úspešnosť v troch z piatich prípadov vyššia ako 95%.

pole markerov	HTC noc	HTC deň	Canon noc	Canon deň	priemer
colours db	76.92%	83.65%	74.04%	76.92%	77.88%
colours	35.57%	33.65%	25.96%	42.31%	34.37%
country db	71.15%	73.08%	75.96%	90.38%	77.64%
country db *	97.12%	96.15%	89.42%	100%	95.67%
country	62.5%	75%	61.54%	64.42%	65.87%
trenčín db	96.15%	100%	89.42%	96.15%	95.43%
trenčín	60.57%	88.46%	71.15%	74.04%	73.56%
vut db	99.04%	100%	99.04%	100%	99.52%
vut	90.38%	94.23%	89.42%	92.31%	91.59%

Tabuľka 5.7: Úspešnosť detekcie markerov pri oprave 4 chýb

Dosiahnuté výsledky sa javia byť dostatočné na použitie farebných polí markerov, tak ako ich predstavuje táto diplomová práca, v aplikáciách využívajúcich rozšírenú realitu. Na ďalšie zvyšovanie úspešnosti odhadu pozície kamery by mohol byť použitý prístup popísaný vyššie. Inou možnosťou je rozšírenie dátovej sady používanej pri tvorbe databázy úspešnosti dvojíc a následné zvýšenie počtu farebných oblastí.

Kapitola 6

Záver

V práci boli popísané základy rozšírenej reality so zameraním sa na algoritmy výpočtu polohy kamery pomocou rôznych typov markerov. V nadväznosti na to boli predstavené základné myšlienky z článku *Uniform Marker Fields*, ktorý sa zaoberá generovaním a detekciou uniformných polí markerov. Boli navrhnuté rozšírenia tohto prístupu, ktoré slúžili ako východiskový bod pre praktickú časť práce.

Bol vytvorený program na generovanie farebných polí markerov na základe obrazovej predlohy. Generovanie prebieha pomocou genetického algoritmu, hodnotiaca funkcia sa zameriava na dodržanie troch podmienok: minimálny počet konfliktných markerov, maximálna podobnosť s obrazovou predlohou a maximálna úspešnosť detekcie. Pomocou vytvoreného nástroja na detekciu je možné určiť úspešnosť detekcie jednotlivých markerov.

Nakoľko výsledky detekcie neboli uspokojivé, bolo po dohode s vedúcim práce rozhodnuté zamerať sa na odhalenie príčin nízkej úspešnosti a na jej zlepšenie. Bola vykonaná štúdia stálosti farieb v procese tlače a snímania ako aj úspešnosti detekcie prechodov pre rôzne dvojice farieb. Získané poznatky sú jednak podrobne diskutované, taktiež ich možno vo forme textových záznamov použiť pre iné aplikácie. Ich využitie sa neobmedzuje iba na oblasť rozšírenej reality, môžu sa uplatniť v obecnej aplikácii využívajúcej farby na kódovanie informácie. Z toho dôvodu predstavujú najväčší prínos tejto diplomovej práce.

Poznatky získané pomocou štúdie boli zahrnuté do hodnotiacej funkcie genetického algoritmu. To malo za následok značné zvýšenie úspešnosti detekcie markerov. S využitím schopnosti opraviť určité množstvo chybných prechodov bola dosiahnutá úspešnosť detekcie, ktorá sa javí ako dostatočná pre použitie v reálnej aplikácii využívajúcej rozšírenú realitu.

Z pohľadu ďalšieho vývoja práce sa javí byť veľmi dôležitým rozširovanie dátovej sady, ktorá bola použitá v štúdiu. S dostatočne rozsiahlou sadou vstupných dát by bolo možné zväčšiť množstvo farebných oblastí a tým pádom spresniť výsledky štúdie. Ich začlenenie do hodnotiacej funkcie genetického algoritmu by sa malo prejaviť zvýšením úspešnosti detekcie markerov.

Literatura

- [1] Azuma, R. T.: A Survey of Augmented Reality. *Presence*, ročník 6, 1997: s. 355–385.
- [2] Burns, J.; Mitchell, C. J.: Coding Schemes for Two-Dimensional Position Sensing. *Institute of Mathematics and Its Applications Conference Series*, 1993.
- [3] Milgram, P.; Kishino, F.: A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, ročník E77-D, č. 12, Prosinec 1994.
- [4] Milgram, P.; Takemura, H.; Utsumi, A.; aj.: Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. In *Proceedings of the SPIE Conference on Telemanipulator and Telepresence Technologies, Proceedings of SPIE*, ročník 2351, Boston, Massachusetts, USA, Listopad 1995, s. 282–292.
- [5] Park, J.; You, S.; Neumann, U.: Natural feature tracking for extendible robust augmented realities. In *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes: placing artificial objects in real scenes*, IWAR '98, Natick, MA, USA: A. K. Peters, Ltd., 1999, ISBN 1-56881-098-9, s. 209–217.
- [6] Szentandrás, I.; Zachariáš, M.; Havel, J.; aj.: Uniform Marker Fields: Camera Localization By Orientable De Bruijn Tori. In *Proceedings of ISMAR*, 2012.
- [7] Tateno, K.; Kitahara, I.; Ohta, Y.: A nested marker for augmented reality. In *ACM SIGGRAPH 2006 Sketches*, SIGGRAPH '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-364-6, doi:10.1145/1179849.1180039.
- [8] WWW stránky: ARToolKit documentation (How does ARToolKit works?). [cit. 2012-01-08].
URL <http://www.hitl.washington.edu/artoolkit/documentation/userarwork.htm>
- [9] WWW stránky: ARToolKit home page. [cit. 2012-01-08].
URL <http://www.hitl.washington.edu/artoolkit/>
- [10] WWW stránky: EyeTap Personal Imaging Lab. [cit. 2012-01-08].
URL <http://eyetap.org/>
- [11] WWW stránky: Google Glass není první ani nejlepší technologie. [cit. 2012-01-08].
URL <http://www.zive.cz/clanky/google-glass-neni-prvni-ani-nejlepsi-technologie/sc-3-a-166980/default.aspx>

- [12] WWW stránky: Home of the Marker Fields. [cit. 2012-01-08].
URL <http://www.fit.vutbr.cz/research/groups/graph/MF/>
- [13] WWW stránky: Vuforia — Augmented Reality Unleashed. [cit. 2012-01-08].
URL <https://www.vuforia.com/>
- [14] Zhang, X.; Fronz, S.; Navab, N.: Visual Marker Detection and Decoding in AR Systems: A Comparative Study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, Washington, DC, USA: IEEE Computer Society, 2002, ISBN 0-7695-1781-1, s. 97–106.
- [15] Zhou, F.; Duh, H. B.-L.; Billinghurst, M.: Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, Washington, DC, USA: IEEE Computer Society, 2008, ISBN 978-1-4244-2840-3, s. 193–202, doi:10.1109/ISMAR.2008.4637362.

Príloha A

Adresárová štruktúra DVD

Príloha popisuje adresárovú štruktúru priloženého DVD do druhej úrovne.

\bin	spustiteľné súbory programov
\colourTester	
\gridDetector	
\MFGenerator	
\transitionDetector	
\doc	dokumentácia zdrojových textov a správa
\doxygen	
sprava.pdf	
\misc	plagát a prezentačné video
plagat.jpg	
video.wmv	
\src	zdrojové súbory programov a správy
\colourTester	
\gridDetector	
\MFGenerator	
\sprava	
\transitionDetector	
\study	textové a grafické výsledky štúdií
\colorPairs	
\singleColor	

Príloha B

Ovládanie programov

B.1 Program MFGenerator

Program *MFGenerator* slúži na generovanie polí markerov. Vstupom programu je obrazová predloha, výstupom je vygenerované pole markerov `markerField.jpg`, vygenerované pole markerov s vyznačenými prechodmi `markerFieldTransitions.jpg` a textový súbor so zoznamom prechodov `transitions.txt`. Váhy `-ws` a `-wd` musia mať v súčte hodnotu 1, v prípade použitia databázy musia tento súčet dosiahnuť všetky tri váhy: `-ws`, `-wd` a `-wdb`.

Parametre programu (povinné parametre sú označené hviezdičkou):

- `-h` Nápoveda k programu
- `-i` <názov> *
 Vstupný obrázok s obrazovou predlohou
- `-r` <šírka><výška>
 Šírka a výška výsledného poľa markerov
- `-p` <n>
 Veľkosť populácie genetického algoritmu
- `-e` <n>
 Počet evolúcií genetického algoritmu
- `-db` <názov>
 Databázový súbor s úspešnosťami detekcie prechodov pre dvojice farebných oblastí
- `-ws` <n>
 Váha čiastkovej hodnotiacej funkcie zabezpečujúcej podobnosť s obrazovou predlohou v intervale $\langle 0.0, 1.0 \rangle$
- `-wd` <n>
 Váha čiastkovej hodnotiacej funkcie zabezpečujúcej detekovateľnosť v intervale $\langle 0.0, 1.0 \rangle$
- `-wdb` <n>
 Váha čiastkovej hodnotiacej funkcie využívajúcej databázu úspešností prechodov dvojíc farebných oblastí v intervale $\langle 0.0, 1.0 \rangle$

B.2 Program gridDetector

Program *gridDetector* detekuje mriežku vo fotografii poľa markerov. Výstupom je zoznam stredov políčok mriežky, ktorý je vypísaný na štandardný výstup. Program v binárnej forme poskytol Ústav počítačové grafiky a multimédií.

Parametre programu (povinné parametre sú označené hviezdíčkou):

- i <názov> *
Vstupná fotografia
- c <n> *
Dĺžka strany štvorcovej mriežky, ktorá bude detekovaná
- s
Grafické zobrazenie výsledku

B.3 Program transitionDetector

Program *transitionDetector* rekonštruuje vygenerované pole markerov, detekuje typy prechodov a vypočíta úspešnosti detekcie. Vstupom je textový súbor so zoznamom prechodov vo vygenerovanom poli markerov, fotografia poľa markerov a textový súbor so stredmi mriežky, ktorá v nej bola detekovaná. Úspešnosti detekcie sú vypísané na štandardný výstup, rekonštruované polia markerov bez vizualizácie, s vizualizáciou všetkých / chybných prechodov sú uložené ako `markerField.jpg`, `markerFieldT.jpg` a `markerFieldET.jpg`.

Parametre programu (povinné parametre sú označené hviezdíčkou):

- h
Nápoveda k programu
- i <názov> *
Vstupná fotografia
- e <názov> *
Textový súbor s prechodmi
- c <názov> *
Textový súbor so stredmi mriežky

B.4 Program colourTester

Program *colourTester* slúži na vytvorenie databázy odchýlok u jednotlivých farieb v procese tlače a snímania alebo úspešností detekcie prechodov pre dvojice farieb (predvolená funkcionálnosť). Vstupom sú dvojice súborov — vygenerované a rekonštruované polia markerov. Každá dvojica je uložená vo zvlášť adresári, ktorého názvom je jeho poradové číslo, a ktorý sa nachádza v adresári `\Data`. Vstupné súbory sú pomenované `gen.jpg` a `rec.jpg`. Databáza je exportovaná do textového súboru s názvom `database.txt`, graf úspešností dvojíc farebných oblastí je uložený ako `graph.jpg`.

Parametre programu (povinné parametre sú označené hviezdikou):

```
-h                Nápoveda k programu
-nDir <n> *        Počet dvojíc vstupných súborov
-nInt <n> *        Počet intervalov na ktoré bude rozdelený každý farebný kanál
-scdb <názov>      Skratka znamená single color database a zapríčiňuje vytvorenie databázy
                   odchýlok pre jednotlivé farby
```